Toward Vehicle-Assisted Cloud Computing for Smartphones

Hongli Zhang, Qiang Zhang, and Xiaojiang Du

***Abstract*—Mobile cloud computing is an emerging technology for facilitating complex application execution on smartphones. Cloud services are utilized not only to speed up the running of mobile applications but to save energy for smartphones as well. In this paper, we propose to combine the vehicular cloud with the infrastructure-based cloud to expand the current available resources for task requests from smartphones. In our proposed architecture, the vehicular cloud acts as a cloud service provider for smartphones. Moreover, we propose a flexible offloading strategy (FOS) to carry out task migration. The vehicular cloud is able to discover and utilize the underutilized resources in vehicles to accomplish application offloading for smartphones. The FOS estimates the efficiency of various cloud service providers based on current resource conditions and then selects the suitable cloud service provider to perform the requested task. Experimental results show that the proposed approach can improve the performance of mobile applications on smartphones in terms of task response time and energy consumption.**

***Index Terms*—Cloudlet, mobile cloud computing, task migration, vehicular cloud.**

I. Introduction

SMARTPHONES have become very popular because they are portable and can run many kinds of applications. However, the portability of smartphones also limits their size and weight; hence, some resources on smartphones are limited, for example, computation resources and storage resources. Computing power and memory capacity of smartphones are gradually growing, but they still cannot meet the computation resource requirements of some mobile applications that are usually computationally intensive. Many complex applications have poor performance when they are performed by smartphones, for instance, image processing, gaming, and so on. Due to the rich resources in the cloud platform, cloud computing is utilized to facilitate efficient application execution on smartphones.

Mobile cloud computing [1] is an emerging technology that integrates cloud computing and mobile computing to enhance the application performance of mobile devices. As the computing capability of smartphones increases, mobile cloud computing can organize and make use of computation resources on surrounding smartphones to form the ad hoc virtual cloud [2]. In addition to the ad hoc virtual cloud, cloud service providers also include a central cloud and a cloudlet that are referred to as the infrastructure-based cloud. Cloudlets [3] are deployed near Wi-Fi access points (APs) and cellular base stations to provide cloud services efficiently and decrease the network delay between mobile users and the central cloud. Vehicles as mobile devices can be also served by mobile cloud computing [4], [5]. For vehicles, cloud service providers include the central cloud, the roadside cloud, and the vehicular cloud [6]. Vehicular cloud is defined as [7] a group of largely autonomous vehicles whose corporate computing, sensing, communication, and physical resources can be coordinated and dynamically allocated to authorized users. Vehicles traveling on the road may meet various events such as traffic congestion and traffic accidents. In such situations, the vehicular cloud has the potential to cooperate with various authorities to solve problems that otherwise cannot be solved efficiently [8]. Vehicular cloud will play an important role in implementing autonomous traffic, vehicle control, and perception systems [9].

To get good application performance of smartphones, application offloading is employed to speed up application execution and save energy for a smartphone [10]. Offloading is a kind of mechanism utilized to alleviate resource constraints of mobile devices by migrating part or all of the tasks corresponding to an application to resource-rich surrogates [11]. In mobile cloud computing, application offloading mainly means that smartphones utilize cloud services to execute some tasks to implement good application performance. Two offloading techniques have been presented, including system-level offloading and method-level offloading. For example, Clone Cloud [12] and Cloudlet [13] are categorized as system-level offloading. In system-level offloading, a cloned image of a smartphone is created by virtual machine technology and maintained in a cloud platform. MAUI [14] is categorized as method-level offloading. In method-level offloading, program partitioning and fine-grained code migration are implemented.

Some application offloading policies have been proposed to decide on whether the tasks should be migrated from a smartphone to a cloud node or be locally executed. However, most research only focuses on the problem of whether the task should be offloaded to a dedicated cloud node such as the central cloud or the cloudlet. When the dedicated cloud node cannot satisfy the offloading requirements, the cloud services become useless for smartphones. Therefore, it is necessary to

Manuscript received April 13, 2015; revised July 9, 2015; accepted September 2, 2015. Date of publication September 18, 2015; date of current version December 14, 2015. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2011CB302605, by the National Natural Science Foundation of China under Grant 61202457, by the National Science Foundation under Grant CNS-1022552 and Grant CNS-1065444, and by the U.S. Army Research Office under Grant WF911NF-14-1-0518. The review of this paper was coordinated by the Guest Editors.

H. Zhang and Q. Zhang are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: zhanghongli@hit.edu.cn; zhangqiang@pact518.hit.edu.cn).

X. Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2015.2480004

integrate various usable resources to enhance the scalability of cloud services for smartphones. Once various computation resources are integrated and utilized, there will be multiple cloud nodes to provide cloud services. When one of the cloud nodes cannot satisfy the offloading requirements, other cloud nodes may be available for cloud services. Thus, the cloud services are still useful for smartphones. In addition, there may be a situation where several cloud nodes satisfy the offloading requirements. Then, how to select an optimal cloud node to execute the application needs to be studied.

In this paper, we propose to combine the vehicular cloud with the fixed central cloud and cloudlet to expand the current available resources for task requests from smartphones. In the proposed cloud computing architecture, vehicular cloud near the cloudlet acts as a cloud service provider for smartphones. When infrastructure-based cloud does not satisfy the offloading requirements, the proposed flexible offloading strategy (FOS) discovers and utilizes the underutilized resources in vehicles to accomplish mobile application offloading. If both the central cloud and cloudlet satisfy the offloading requirements, the FOS selects the optimal cloud node (central cloud or cloudlet) according to the energy consumption of the smartphone and the task response time.

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents the system model and states the problem. Section IV describes the proposed flexible application offloading strategy for smartphones. Section V presents the numerical performance evaluation, and Section VI concludes this paper.

II. Related Work

Mobile application offloading and the corresponding offloading strategy have been investigated by researchers. Currently, the central cloud and cloudlet are cloud service providers most commonly used by smartphone users. Researchers have mainly focussed on the analysis of energy consumption and data transmission time when a smartphone migrates a computationally intensive task to the cloud side. Our work considers the tasks corresponding to different cases of data communication and computation workload. Furthermore, we analyze the offloading performance corresponding to the central cloud and cloudlet based on the related work. Finally, we select the suitable cloud node to perform the requested task using our proposed offloading strategy. Hence, we first introduce the related work on the analysis of offloading performance and the corresponding offloading strategy. Then, we give the comparison between our work and current methods.

An analysis [15] on the energy saving of cloud computing for mobile users is proposed. The researchers point out that cloud computing can potentially save energy for mobile users. However, not all applications are energy efficient when migrated to the cloud. The corresponding model contains the mobile device and a central cloud. The work focuses on the energy analysis for mobile application offloading. When a smartphone uses application offloading, its energy consumption is mostly caused by wireless transmission. Hence, the time to transmit data is a key factor for energy consumption. Moreover, the transmission time is decided by the data size and the bandwidth. The work argues that when the data size is large, offloading may not save energy. However, the problem of how to process the data-intensive task was not discussed. Moreover, the performance metric of task response time was not analyzed.

An optimization framework for energy-optimal application execution [16] is proposed to optimally execute mobile applications in either the mobile device or the cloud clone. The corresponding model contains the mobile phone and a central cloud. The proposed method can calculate the minimum computation energy consumption and the minimum transmission energy consumption with a time deadline on the mobile phone. The proposed framework decides on where to execute the application according to the energy consumption conditions of local execution and cloud clone. However, when the task is data intensive, the data transmission time may exceed the time deadline due to wide area network (WAN) delay. The work did not consider this situation.

Cloudlet is designed to obtain the resource benefits of cloud computing without incurring WAN delays and jitter corresponding to the central cloud [13]. A cloudlet can be viewed as a cluster of multicore computers with Internet connectivity and a high-bandwidth wireless local area network (WLAN), and the corresponding communication range is one hop. The computation and storage resources in the cloudlet can be utilized to perform application offloading by nearby mobile computers. However, if no cloudlet is available nearby, a mobile user cannot migrate the data-intensive task efficiently. The performance metric of energy consumption of smartphones was not analyzed. Moreover, when many users have requests, the resources in a cloudlet may be insufficient. This situation was not discussed.

A unified elastic computing platform [2] is proposed to support application offloading for mobile devices and reduce energy consumption of smartphones based on [16]. The proposed model consists of an infrastructure-based cloud and an ad hoc virtual cloud formed by a cluster of smartphones. The researchers presented an offloading policy to decide on where each task of the application should be executed (i.e., on the standalone smartphone, in the ad hoc virtual cloud, or in the infrastructure-based cloud). The offloading performance of the ad hoc virtual cloud was not analyzed and discussed. There is no comparison between the infrastructure-based cloud and the ad hoc virtual cloud in terms of energy consumption of smartphones and task response time. The offloading policy only implemented the selection between the infrastructure-based cloud and local execution.

Most of the current work considered a single cloud service provider for smartphones. It is very possible that the single cloud service provider becomes useless because its resource conditions are unsuitable for the requested task. For example, the central cloud is unsuitable for a data-intensive task, considering the low data transmission rate of a WAN. In particular, if the data-intensive task belongs to a real-time application, the transmission delay of a WAN is possibly unacceptable for users. Although the cloudlet makes cloud services better for the data-intensive task, it also has limitations. Since the computation and storage resources in the cloudlet are limited, when a large number of users have requests, it is possible that the cloudlet does not have sufficient resources to provide

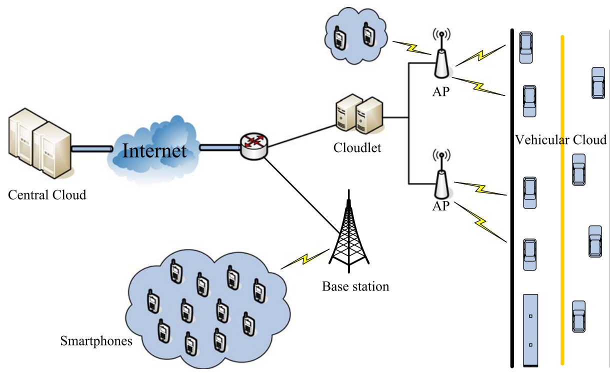

Fig. 1. VACC architecture.

cloud services. Moreover, the one-hop communication range of the cloudlet is also limited. To efficiently offload the tasks corresponding to different cases of data communication and computation workload to a suitable cloud service provider, our work analyzes the offloading performance of various cloud nodes in terms of task response time and energy consumption. To relieve the limitation of the cloudlet, our work establishes a communication model to expand the communication range of the cloudlet and employs the resources in the vehicular cloud to finish mobile application offloading for smartphones. Currently, most research studies aim to perform complex applications on vehicles and improve the intelligent transport system using the vehicular cloud [4]–[8]. However, for an application on a vehicle, the performance of the application executed by the vehicular cloud was not analyzed. Our work analyzes the offloading performance of the vehicular cloud for smartphones. Based on the mobility and resource conditions of worker nodes in the vehicular cloud, we design an algorithm to select a reliable worker node for the requested task. When a smartphone user performs task migration, our work considers both the characteristics of the requested task and the resource conditions of various cloud nodes to select the suitable cloud node. Compared with the current work, our work enhances the scalability and feasibility of cloud services for smartphones in the aspect of task migration.

III. System Model and Problem Statement

Fig. 1 demonstrates the proposed vehicle-assisted cloud computing (VACC) architecture. VACC contains a central cloud, a cloudlet, and a vehicular cloud. Vehicular cloud is the combination of cloud computing and vehicular networks. The original aim of vehicular cloud is to exploit the underutilized resources in vehicles to provide better support for complex applications and services in vehicles and transport systems. VACC transforms the conventional vehicular cloud model and aims to exploit the underutilized resources in vehicles to increase the available resources of cloud services for smartphones. In VACC smartphones, the vehicular cloud and the cloudlet are in a metropolitan area network (MAN). The vehicular cloud is connected to a cloudlet through a WLAN created by a Wi-Fi AP. Smartphones can connect cloudlets through cellular third-generation (3G)/fourth-generation (4G) networks or Wi-Fi APs; hence, smartphones can connect to the vehicular cloud by the cloudlet. Apart from the aforementioned connection method, a smartphone on a bus can directly connect to the bus through Wi-Fi technology. Thus, in VACC, smartphones can access three types of cloud nodes, including the central cloud, the cloudlet, and the vehicular cloud.

Cloud nodes in VACC are heterogeneous; hence, their corresponding cloud services are different on resource conditions, including computing power, memory capacity, network bandwidth, network connectivity, and so on. Generally, a central cloud has rich computation and storage resources, but it has a low data transmission rate because it transfers data through a WAN. Cloudlet has limited computation and storage resources, but it has a high data transmission rate since it uses a WLAN or a MAN for data transmission. Meanwhile, a cloudlet as a public infrastructure has a high resource utilization rate due to its convenience and a large number of service requests. Each node in a vehicular cloud has limited computation and storage resources, but the total computation and storage resources of a vehicular cloud are rich. In addition, the resources in a vehicular cloud are dispersed, and the resource utilization rate is low. However, the vehicular cloud has a high data transmission rate for smartphones because it employs WLAN or MAN communications. According to the resource features of various cloud nodes, we give the assumptions as follows. The computation and storage resources in the central cloud are unlimited, even with heavy load of offloading requests from many users. The computation and storage resources in the cloudlet are limited. When many users have requests, the resources in the cloudlet may be insufficient.

VACC adopts the system-level offloading technique (i.e., virtual machine) for the central cloud and the cloudlet. We consider that, normally, the assigned virtual machine for a smartphone cannot change its total amount of computation and storage resources when it is working. For the vehicular cloud, VACC adopts the method-level offloading technique considering moderate computation and storage resources in vehicles. For a mobile application, it may contain one or more tasks. This paper only discusses the case in which an application only contains one task, and the corresponding task cannot be divided into several parts. Considering that only a task needs to be migrated for a requested application, this paper only selects a worker node (i.e., a vehicle) in the vehicular cloud to perform the requested task for smartphones. We will discuss the case in which an application contains multiple tasks in our future work, and these tasks will be divided and allocated to several worker nodes in the vehicular cloud to implement distributed and parallel computation.

Based on the given model and assumptions, we focus on the following problems: which cloud node the requested task should be migrated to and how to find a reliable worker node to execute the task when the vehicular cloud has been selected as the cloud service provider. To solve the first problem, the offloading performance of the central cloud and the cloudlet should be analyzed. If both the central cloud and the cloudlet satisfy the offloading requirements, the problem becomes the selection between the central cloud and the cloudlet. If both the central cloud and the cloudlet cannot satisfy the offloading requirements, the vehicular cloud should be selected as the cloud service provider. Compared with the first problem, the second problem is more complex. To solve the second problem, several factors should be considered. Due to the mobility of worker nodes in the vehicular cloud, the connection between a smartphone user and the vehicular cloud is not stable. If the duration of the connection is too short, the data transmission of the requested task possibly cannot be finished. Hence, we should select the worker node with longer connected time. However, the worker node with longer connected time may not have sufficient resources. If the cloudlet uses unicast to discover each worker node and collect the corresponding status information, the total delay of multiple-worker-node discovery is larger. Therefore, to find a qualified node, multiple worker nodes need to be discovered by the cloudlet using multicast. The time to discover multiple worker nodes impacts the task response time for smartphone users. If we discover and collect all nodes in the vehicular cloud, the probability of finding a qualified node will be higher. However, the corresponding task response time may not satisfy the offloading requirement. Hence, we try to find a reliable worker node within a certain amount of worker nodes in the vehicular cloud.

IV. Flexible Offloading Strategy for Smartphones

In VACC, the vehicular cloud plays an assistant role in cloud computing for smartphones. The vehicular cloud is exploited to relieve the limitation of the infrastructure-based cloud, including the WAN delay of the central cloud and the limited available resources of the cloudlet. Hence, the offload scenarios in VACC can be divided into two categories. In the first category, either the central cloud or the cloudlet can satisfy the offloading requirements, and the requested task is migrated to one of them based on the offloading performance. In the second category, both the central cloud and the cloudlet cannot satisfy the offloading requirements, and the task is migrated to the vehicular cloud. For the central cloud, the FOS calculates the energy consumption values of a smartphone and task response time. For the cloudlet, the FOS judges whether the current available resources in the cloudlet satisfy the resource requirement of the requested task. If yes, the FOS continues to calculate the energy consumption values of a smartphone and task response time corresponding to the cloudlet. Then, the FOS determines whether the central cloud and the cloudlet satisfy the requirements of task response time and energy consumption. If both the central cloud and the cloudlet satisfy the offloading requirements, the FOS will select the optimal cloud node (central cloud or cloudlet) to execute the task. When only the central cloud or the cloudlet satisfies the offloading requirements, the task will be executed by the corresponding cloud node. If both the central cloud and the cloudlet cannot satisfy the offloading requirements, the FOS will discover and utilize the vehicular cloud to accomplish application offloading.

The vehicular cloud in VACC contains two situations. A request from a smartphone should be processed in the corresponding situation. In the first situation, the user is not on a bus, but rather, he/she may be walking on a street. His/her smartphone needs to connect to the cloudlet to discover and utilize a vehicular cloud. In the second situation, the user is on a bus, and the smartphone can directly connect to the vehicle through Wi-Fi technology. In the first situation, the network between the smartphone and the vehicular cloud is intermittently connected; hence, VACC needs to select a reliable worker node (i.e., the lifetime of the connection between the smartphone and the requested vehicle is long enough to receive the task) to implement task migration. Meanwhile, a reliable worker node also needs to satisfy the offloading requirements. In the second situation, the network connection between the smartphone and the vehicle is relatively stable. If the vehicle satisfies both the offloading requirements and the connection requirement, VACC will employ it to execute the task.

In the following sections, we analyze the offloading performance of the central cloud and the cloudlet. Moreover, we design an algorithm to select the reliable worker node in the vehicular cloud. Table I shows the notation definition in the FOS.

TABLE I
Notation Definition in FOS

Symbol	Description
P_u^c	Power of processor in smartphone
T	Required response time
P_{tx}^c/P_{rx}^c	Transmission / receiving power of smartphone using cellular 3G/4G networks
P_{tx}^w/P_{rx}^w	Transmission / receiving power of smartphone using Wi-Fi technology
C_u	The number of CPU cycles corresponding to the computing workload of the task
$S_u/S_{cc}/S_{cl}/S_{vc}^i$	CPU clock frequency of smartphone / central cloud / cloudlet / worker node i in vehicular cloud
D_u/D_r	Input / output size of the task
B_{tx}^c/B_{rx}^c	Upload / download rate of smartphone using central cloud
B_{tx}^l/B_{rx}^l	Upload / download rate of smartphone using cloudlet
B_{tx}^v/B_{rx}^v	Upload / download rate of smartphone using vehicular cloud

A. Offloading Performance Analysis of the Central Cloud and the Cloudlet

The offloading performance of the central cloud and the cloudlet can be calculated as follows. The energy consumption by local execution is

$$E_u = P_u^c \times \frac{C_u}{S_u}. \tag{1}$$

The energy consumption of the smartphone corresponding to the central cloud is

$$E_{cc} = P_{tx}^c \times \frac{D_u}{B_{tx}^c} + P_{rx}^c \times \frac{D_r}{B_{rx}^c}. \tag{2}$$

The energy consumption of the smartphone corresponding to the cloudlet is

$$E_{cl} = P_{tx}^c \times \frac{D_u}{B_{tx}^l} + P_{rx}^c \times \frac{D_r}{B_{rx}^l}. \tag{3}$$

The task response time corresponding to the central cloud is

$$T_{cc} = \frac{D_u}{B_{tx}^c} + \frac{C_u}{S_{cc}} + \frac{D_r}{B_{rx}^c}. \quad (4)$$

The task response time corresponding to the cloudlet is

$$T_{cl} = \frac{D_u}{B_{tx}^l} + \frac{C_u}{S_{cl}} + \frac{D_r}{B_{rx}^l}. \quad (5)$$

If the central cloud can satisfy the conditions of $E_{cc} < E_u$ and $T_{cc} \leq T$ at the same time, it is considered to be available to perform application offloading.

If the current available resources in the cloudlet are sufficient for the requested task and the cloudlet can satisfy the conditions of $E_{cl} < E_u$ and $T_{cl} \leq T$ at the same time, it is considered to be available to perform application offloading.

When there are two available cloud nodes (i.e., central cloud and cloudlet), the FOS needs to select the optimal cloud node. If $E_{cc} \leq E_{cl}$, then the central cloud is selected to execute the requested task. Otherwise, the optimal cloud node is the cloudlet. When there is only one available cloud node (i.e., central cloud or cloudlet), the corresponding cloud node is selected.

When both the central cloud and the cloudlet are unavailable for the requested task, the FOS will discover and utilize the vehicular cloud to execute the task.

B. Worker Node Selection in Vehicular Cloud

In the first situation of the vehicular cloud, first, a smartphone sends a request to the cloudlet and then utilizes the AP connected with the cloudlet to discover the vehicular cloud. The request from a smartphone contains the energy consumption E_u, the required response time T, the required memory size M, and the set $\{D_u, C_u, D_r\}$. The total number of nodes in the vehicular cloud is n. The cloudlet randomly selects m vehicles in the communication range of APs as candidate worker nodes. Then, some status information of m vehicles is transferred to the cloudlet. The status information of vehicle i consists of the current speed V_i, the direction θ_i, the available memory M_v^i, the CPU clock frequency S_{vc}^i, and the coordinate (X_i, Y_i). After the status information of a vehicle is received by the cloudlet, the corresponding connection is successfully established.

Next, the cloudlet estimates the duration of the connection between the smartphone and the vehicular cloud. Because the network connection between the smartphone and the cloudlet is stable, the previously estimated duration is equal to the duration of the connection between the cloudlet and the candidate worker node in the vehicular cloud. For vehicle i, the estimated duration can be obtained by the following equation:

$$R = \sqrt{(X_i + t_i \times V_i^x - X_s)^2 + (Y_i + t_i \times V_i^y - Y_s)^2}. \quad (6)$$

In (6), $V_i^x = (V_i + V_m) \sin\theta_i/2$, and $V_i^y = (V_i + V_m) \cos\theta_i/2$. V_m represents the permitted maximum speed on the road. (X_s, Y_s) denotes the coordinate of the AP. R denotes the communication range of the AP. We use the average value of the current speed and maximum speed to calculate time t_i. When a vehicle decelerates or keeps the current speed, the actual duration of the connection will be greater than the estimated value; hence, this situation does not destroy the normal task migration. When a vehicle accelerates, the actual duration of the connection is close to the estimated value. If the task is executed by a worker node in the vehicular cloud, the corresponding energy consumption of the smartphone is

$$E_{vc} = P_{tx}^c \times \frac{D_u}{B_{tx}^v} + P_{rx}^c \times \frac{D_r}{B_{rx}^v}. \quad (7)$$

Equation (7) denotes the energy consumption when the smartphone uses 3G/4G networks to connect to the vehicular cloud. If the smartphone uses the Wi-Fi AP to connect to the nearby vehicular cloud, the corresponding energy consumption is

$$E_{vc} = P_{tx}^w \times \frac{D_u}{B_{tx}^v} + P_{rx}^w \times \frac{D_r}{B_{rx}^v}. \quad (8)$$

The task response time corresponding to vehicle i can be calculated as follows:

$$T_{vc}^i = \frac{D_u}{B_{tx}^v} + \frac{C_u}{S_{vc}^i} + \frac{D_r}{B_{rx}^v} + m \times T_v \quad (9)$$

where T_v denotes the time to establish the connection between a vehicle and the cloudlet.

Based on the offloading requirements and the estimated duration of the connection between the vehicular cloud and the smartphone, we design Algorithm 1 to select a reliable worker node, and the selected worker node will execute the task. In the first situation of the vehicular cloud, if the time to transfer the input data to a worker node is less than the estimated duration of the connection, we consider that the worker node satisfies the connection requirement. After the task has been finished, if the worker node is out of the communication range of the AP, the result can be transferred to the cloudlet through multihop vehicular networks. Then, the cloudlet can return the result to the smartphone through 3G/4G networks or Wi-Fi technology. The values of energy consumption and task response time in Algorithm 1 can be calculated by (7)–(9). For brevity, the equations do not appear in Algorithm 1.

Algorithm 1

INPUT: $W : \{w_1, w_2, \ldots, w_m\}$, $S : \{t_1, t_2, \ldots, t_m\}$, T, E_u, D_u, C_u, D_r, $S_{vc} : \{S_{vc}^1, S_{vc}^2, \ldots, S_{vc}^m\}$, $M_v : \{M_v^1, M_v^2, \ldots, M_v^m\}$

for (i = 1; i <= m − 1; i + +)
  $T_s = t_i$;
  k = 0;
  for (j = 1; j <= m − i; j + +)
    if $(T_s < t_{j+1})$
      $T_s = t_{j+1}$;
      k = j + 1;
    endif
  endfor
if $(T_s > t_i)$
    switch (t_i, t_k);
    switch (w_i, w_k);
    switch (S_{vc}^i, S_{vc}^k);
    switch (M_v^i, M_k^v);

```
    endif
    if (M < M_v^i && E_vc < E_u && T_vc^i < T && (D_u/B_tx^v) < t_i)
        node w_i is released as the reliable worker node;
        return;
    endif
if (i = m − 1)
    if (M < M_v^m && < E_u && T_vc^m ≤ T && (D_u/B_tx^v) < t_m)
node w_m is selected as the reliable worker node;
        return;
    endif
endif
endfor
```

The set W denotes the identities corresponding to m candidate worker nodes. The set S denotes the estimated durations of the connections corresponding to candidate worker nodes. Algorithm 1 sorts candidate worker nodes for $(m-1)$ rounds, and the worker node with the longest connection duration is found after the first round. Then, Algorithm 1 judges whether the current worker node satisfies the offloading requirements and the connection requirement. If it does, the current worker node is selected as the reliable worker node; otherwise, Algorithm 1 continues to sort and judge whether the next node satisfies the required conditions. If none of the m candidate worker nodes satisfy the required conditions, then the user can send a request to the cloudlet for vehicular cloud services again. When the cloudlet receives the same request for the second time, it will randomly select m vehicles, except the selected vehicles at the first time. Then, Algorithm 1 will be performed again.

In the second situation of the vehicular cloud (i.e., a user is on a bus), at first, the smartphone sends a request to the current bus x, and the request message is the same as that in the first situation. Then, the FOS judges whether the current bus x satisfies the offloading requirements. Since the smartphone is directly connected to the current bus x through Wi-Fi technology, the corresponding energy consumption of the smartphone is

$$E_{vc} = P_{tx}^{w} \times \frac{D_u}{B_{tx}^{v}} + P_{rx}^{w} \times \frac{D_r}{B_{rx}^{v}} + E_v. \tag{10}$$

The task response time corresponding to the current bus x is

$$T_{vc}^{x} = \frac{D_u}{B_{tx}^{v}} + \frac{C_u}{S_{vc}^{x}} + \frac{D_r}{B_{rx}^{v}} + T_v. \tag{11}$$

In (10), E_v denotes the energy consumption of the smartphone corresponding to the connection established between the smartphone and bus x. In addition, the user needs to offer the estimated duration t_x. If bus x satisfies both the offloading requirements and the condition of $T_{vc}^{x} < t_x$, the task will be executed by bus x. Otherwise, the smartphone will send a request to the cloudlet for the vehicular cloud services in the first situation.

The time complexity of Algorithm 1 is $O(m^2)$ smaller than $O(n^2)$. The main functionality of Algorithm 1 is to select a worker node that satisfies the offloading requirements and the connection requirement for the requested task. Algorithm 1 is able to find a reliable worker node to execute the requested task if there is one or more qualified worker nodes in randomly selected m vehicles, but it also has limitations. It is possible that the there is no qualified worker node in randomly selected m vehicles. Then, the algorithm can run again to find a reliable worker node. However, the time to discover m vehicles at the first time will increase the task response time. Hence, if the algorithm has run several times and cannot find a reliable worker node, it will become useless due to the accumulated response time although a qualified worker node appears next time.

TABLE II
SYSTEM PARAMETERS

Symbol	Value
P_u^c / M	320mW / 25MB
M_l / M_v^i	35MB / 35MB in the first case 15MB / 35MB in the second case 35MB / 35MB in the third case 15MB / 35MB in the fourth case
P_{tx}^c / P_{rx}^c	150mW / 30mW
S_u / S_{cc} / S_{cl} / S_{vc}	(1)500MHz / 5000MHz / 1500MHz / 1200MHz (2)500MHz / 1000MHz / 2000MHz / 1200MHz
B_{tx}^c / B_{rx}^c	800kpbs / 1Mbps for 3G networks
B_{tx}^l / B_{rx}^l	2Mpbs / 3Mbps for 3G networks
B_{tx}^v / B_{rx}^v	2Mpbs / 3Mbps for 3G networks
m / T_v	3 / 0.006s

V. PERFORMANCE EVALUATION

A. Experimental Setting

Our experiments are the theoretical performance evaluation based on our numerical calculations. To implement better theoretical performance evaluation, we need to select the suitable parameters that conform to the practical situations. Generally, if a user wants to use the central cloud, first, he/she needs to buy some resources in the central cloud according to his/her demands. Therefore, if the user usually has medium or low demands, the corresponding resources (e.g., computing power and memory capability) may be moderate or even limited such as a common personal computer. In this situation, the user's smartphone acts as a thin client, and the central cloud acts as a personal computer controlled by the smartphone. If the user often has high demands, then the corresponding resources should be more powerful. Of course, buying powerful resources requires bigger expenses. Hence, in our experiments, we not only select the parameters corresponding to high performance but also use the parameters of medium and low performance for the central cloud. For the networking parameters, we consider that the WAN has a low data transmission rate and that the link between the smartphone and the base station has a high data transmission rate by 3G communication technology.

We set the same CPU clock frequency and available memory space for each candidate worker node in the vehicular cloud, and we assume that there exists one or more available worker

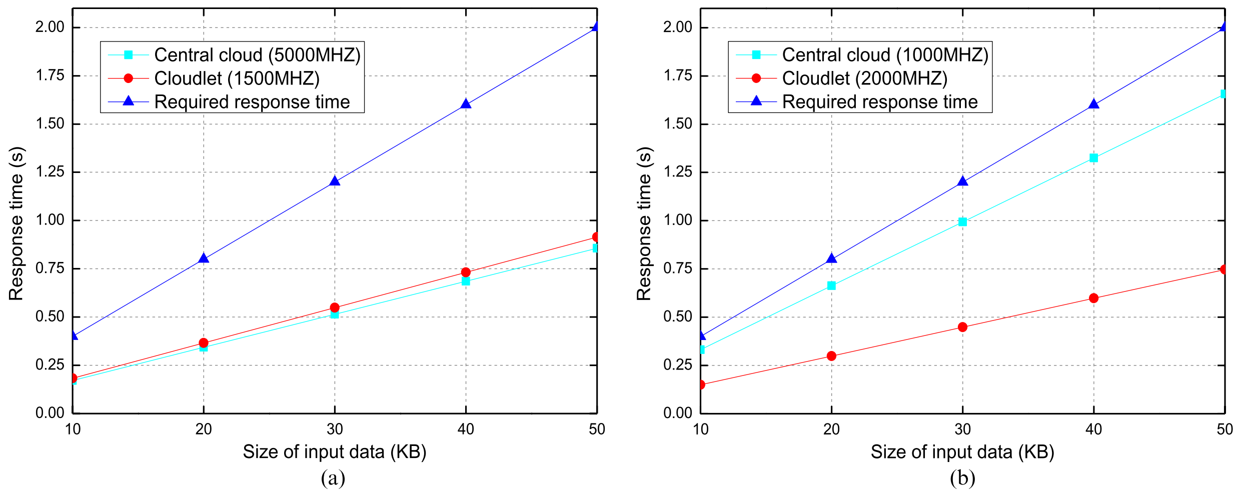

Fig. 2. Task response time evaluation for case 1. (a) Central cloud is powerful. (b) Cloudlet is powerful.

nodes. In the theoretical performance evaluation, we do not consider the mobility of vehicles when the reliable worker node performs the requested task, and we mainly test the computation and communication performance of the vehicular cloud services.

We refer to the system parameters in [17] and [18] to evaluate the theoretical performance. Table II shows the specific parameters in our experiments. M_l denotes the current available memory space of the cloudlet. We test four cases for various requested tasks with different workload and resource conditions of cloud nodes. In each case, different computing capabilities of the central cloud are evaluated. Meanwhile, we test the scenario where the resources in the cloudlet are insufficient for the requested task. For the vehicular cloud, we mainly test the performance using 3G networks.

Case 1: $C_u = (200, 400, 600, 800, 1000)$ M cycles. $D_u = (10, 20, 30, 40, 50)$ KB. $D_r = (4, 8, 12, 16, 20)$ KB. $T = (0.4, 0.8, 0.12, 0.16, 0.20)$ s.

Case 2: $C_u = (100, 200, 300, 400, 500)$ M cycles. $D_u = (15, 30, 45, 60, 75)$ KB. $D_r = (5, 10, 15, 20, 25)$ KB. $T = (0.2, 0.4, 0.6, 0.8, 1.0)$ s.

Case 3: $C_u = (100, 200, 300, 400, 500)$ M cycles. $D_u = (100, 200, 300, 400, 500)$ KB. $D_r = (40, 80, 120, 160, 200)$ KB. $T = (1.0, 2.0, 3.0, 4.0, 5.0)$ s.

Case 4: $C_u = (200, 400, 600, 800, 1000)$ M cycles. $D_u = (200, 400, 600, 800, 1000)$ KB. $D_r = (40, 80, 120, 160, 200)$ KB. $T = (2.0, 4.0, 6.0, 8.0, 10.0)$ s.

B. Experimental Results

The theoretical performance of the FOS corresponding to four cases is shown in Figs. 2–7. In case 1, the workload of data transmission is relatively light. As shown in Figs. 2 and 3, both the central cloud and the cloudlet have a shorter response time than the required time. When the CPU of the central cloud is more powerful than that of the cloudlet, the central cloud has a shorter response time. However, when the cloudlet's CPU is more powerful, the cloudlet is more efficient in terms

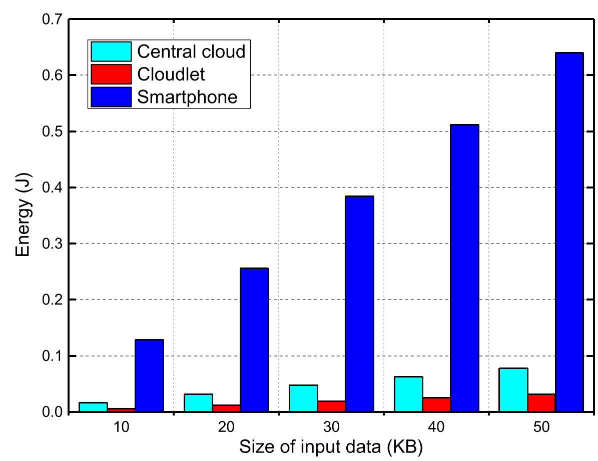

Fig. 3. Energy consumption evaluation for case 1.

of response time. The cloudlet has a high data transmission rate; hence, the transmission delay is shorter, and the energy consumption of the smartphone is less than that of the central cloud. Finally, the FOS selects the cloudlet to execute the task.

In case 2, the computing workload is lighter than in case 1. The available memory space of the cloudlet is not sufficient. As shown in Fig. 4, the vehicular cloud has a shorter response time than the required time. When the CPU of the central cloud is more powerful, the central cloud has a shorter response time, but it is still longer than the required time. Both the central cloud and the vehicular cloud have less energy consumption than the local execution. However, the vehicular cloud has a high data transmission rate; hence, the transmission delay is shorter, and the energy consumption is less than that of the central cloud. Finally, the FOS selects the vehicular cloud to execute the task.

In case 3, the workload of data transmission is heavier than in case 1, which has a heavier computing workload. As shown in Figs. 5 and 6, the cloudlet has a shorter response time than the required time. The central cloud cannot satisfy the offloading requirements in terms of response time and energy consumption. Finally, the FOS selects the cloudlet to execute the task.

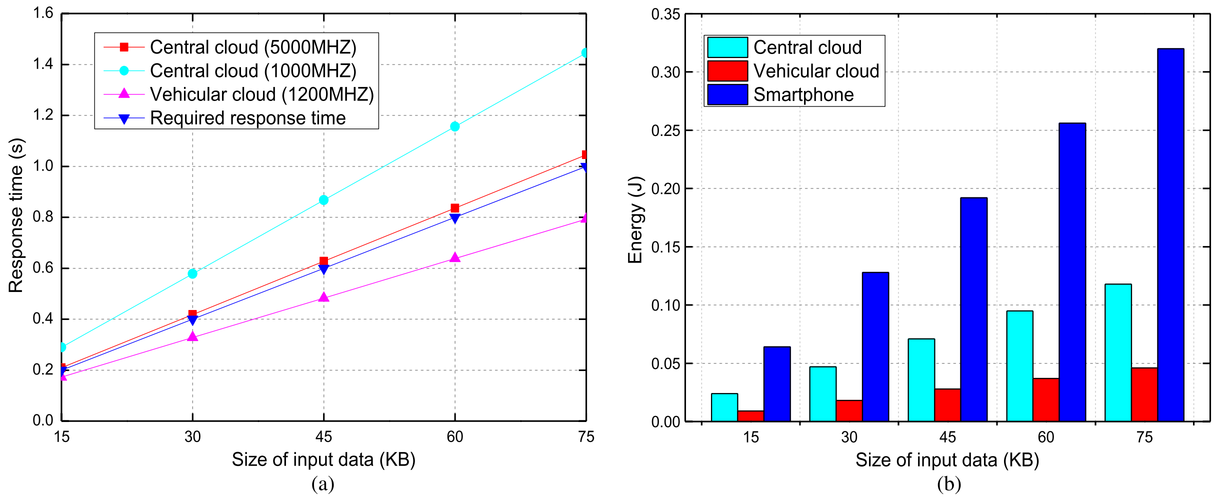

Fig. 4. Performance evaluation for case 2. (a) Task response time. (b) Energy consumption of the smartphone.

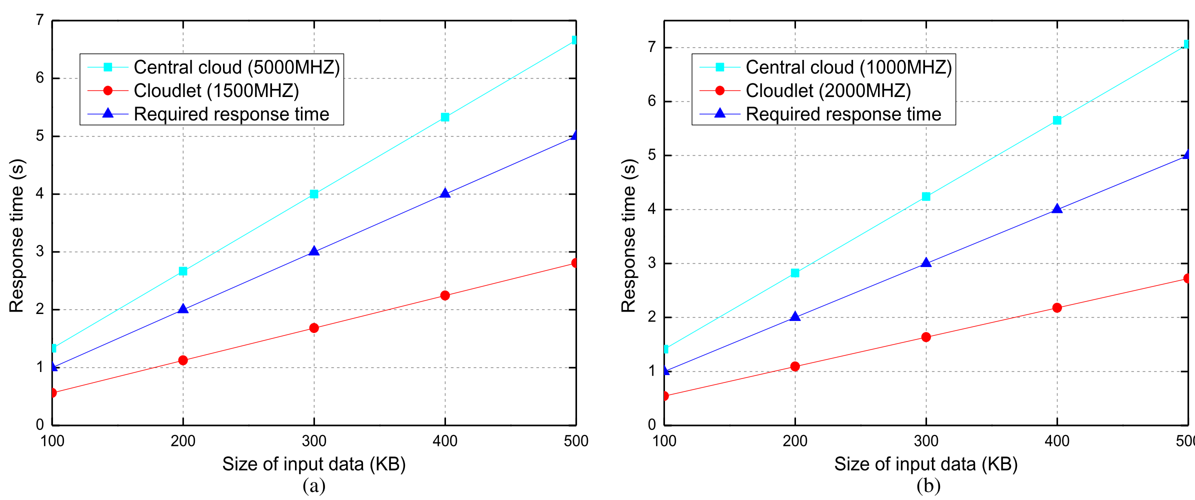

Fig. 5. Task response time evaluation for case 3. (a) Central cloud is powerful. (b) Cloudlet is powerful.

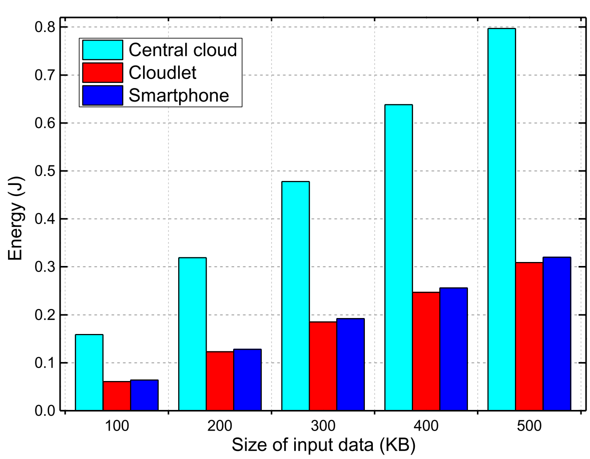

Fig. 6. Energy consumption evaluation for case 3.

In case 4, the workload of data transmission is heavier than in case 3. The available memory space of the cloudlet is not sufficient. As shown in Fig. 7, the vehicular cloud has a shorter response time than the required time, but the response time of the central cloud is longer than the required time due to the low data transmission rate. Meanwhile, the central cloud cannot satisfy the offloading requirements in terms of energy consumption. The vehicular cloud has less energy consumption than the local execution. Finally, the FOS selects the vehicular cloud to execute the task.

VI. Conclusion

In this paper, we have proposed combining the vehicular cloud with the fixed central cloud and the cloudlet to expand the current available resources for task requests from smartphones. In the proposed VACC, the vehicular cloud acts as a cloud service provider for smartphones. To gain the energy saving of smartphones and accomplish tasks in the required time, an FOS is proposed. If both the central cloud and the cloudlet do not satisfy the offloading requirements, then the FOS utilizes the underutilized resources in vehicles to accomplish application offloading. If both the central cloud and the cloudlet satisfy the offloading requirements, the FOS selects the optimal cloud node (central cloud or cloudlet) to execute the task. Numerical experimental results show that the FOS can employ the vehicular cloud to accomplish application offloading for smartphones

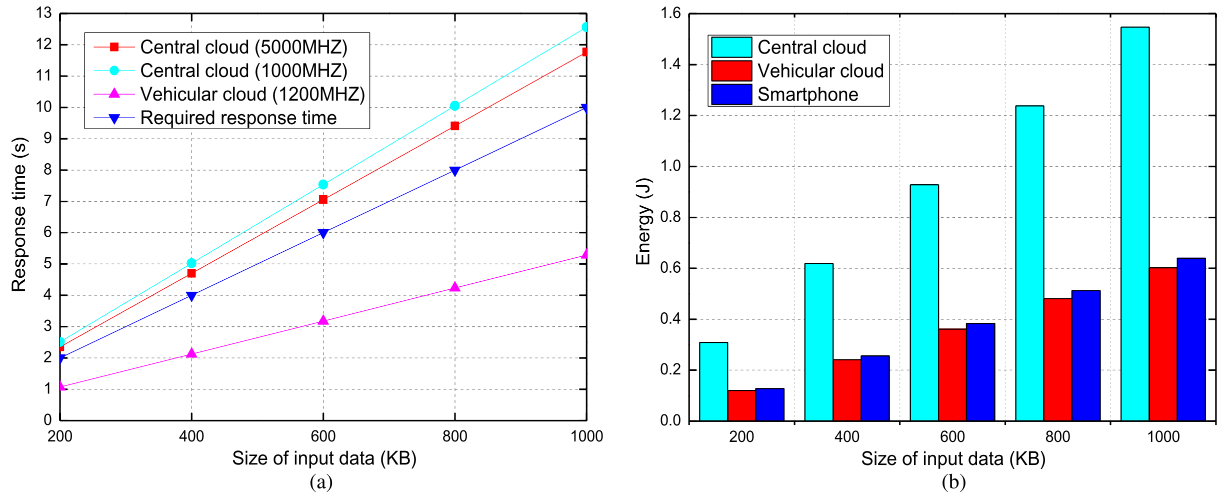

Fig. 7. Performance evaluation for case 4. (a) Task response time. (b) Energy consumption of the smartphone.

when the infrastructure-based cloud does not satisfy the offloading requirements. Therefore, energy savings can be achieved smartphones.

REFERENCES

[1] D. Huang, T. Xing, and H. Wu, "Mobile cloud computing service models: A user-centric approach," *IEEE Netw.*, vol. 27, no. 5, pp. 6–11, Sep./Oct. 2013.

[2] W. Zhang, Y. Wen, J. Wu, and H. Li, "Toward a unified elastic computing platform for smartphones with cloud support," *IEEE Netw.*, vol. 27, no. 5, pp. 34–40, Sep./Oct. 2013.

[3] M. Felemban, S. Basalamah, and A. Ghafoor, "A distributed cloud architecture for mobile multimedia services," *IEEE Netw.*, vol. 27, no. 5, pp. 20–27, Sep./Oct. 2013.

[4] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: Merging VANET with cloud computing," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Technol. Sci.*, 2012, pp. 606–609.

[5] M. Gerla, "Vehicular cloud computing," in *Proc. 11th Annu. Med-Hoc-Net*, 2012, pp. 152–155.

[6] R. Yu, Y. Zhang, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *IEEE Netw.*, vol. 27, no. 5, pp. 48–55, Sep./Oct. 2013.

[7] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *ICST Trans. Mobile Commun. Appl.*, vol. 11, no. 7–9, pp. 1–11, 2011.

[8] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *Int. J. Pervasive Comput. Commun.*, vol. 7, no. 1, pp. 7–21, 2011.

[9] M. Whaiduzzaman, M. Sookhak, A. Gani, R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, no. 1, pp. 325–344, Apr. 2014.

[10] E. Tilevich and Y. Kwon, "Cloud-based execution to improve mobile application energy efficiency," *Computer*, vol. 47, no. 1, pp. 75–77, Jan. 2014.

[11] S. Ou, K. Yang, and A. Liotta, "An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems," in *Proc. IEEE 4th Annu. Int. Conf. Pervasive Comput. Commun.*, Mar. 2006, Pisa, Italy, pp. 116–125.

[12] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, New York, NY, USA, 2011, pp. 301–314.

[13] M. Satyanarayanan, R. C. P. Bahl, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.

[14] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 49–62.

[15] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[16] Y. WenW. Zhang, K. Guan, D. Kilper, and H. Luo, "Energy-optimal execution policy for a cloud-assisted mobile application platform," Nanyang Technol. Univ., Singapore, Tech. Rep., Sep. 2011.

[17] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[18] W. Zhang, Y. Wen, J. Wu, and D. Wu, "Energy-efficient scheduling policy for collaborative execution in mobile cloud computing," in *Proc. IEEE Infocom*, 2013, pp. 190–94.

Hongli Zhang received the B.S. degree in computer science from Sichuan University, Chengdu, China, in 1994 and the Ph.D. degree in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 1999.

She is a Professor with the Department of Computer Science and Technology, HIT, and the Vice Director of the National Computer Information Content Security Key Laboratory. Her research interests include network and information security, network measurement, and cloud computing.

Qiang Zhang received the B.E. degree in information security from Harbin Institute of Technology (HIT), Harbin, China, in 2009 and the M.S. degree from Harbin Engineering University in 2012. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, HIT.

His research interests include cloud computing and wireless networks.

Xiaojiang Du received the B.E. degree from Tsinghua University, Beijing, China, in 1996 and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 2002 and 2003, respectively, all in electrical engineering.

He is currently an Associate Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. He has published over 160 journal and conference papers in his areas of interest. His research interests include security, systems, wireless networks, and computer networks.

Dr. Du has been awarded more than $4M in research grants from the National Science Foundation, the Army Research Office, and the Air Force Research Laboratory. He serves on the Editorial Board of four international journals.