

## Research Article

# A Safety Resource Allocation Mechanism against Connection Fault for Vehicular Cloud Computing

Tianpeng Ye,<sup>1</sup> Zhou Su,<sup>2</sup> Jun Wu,<sup>1</sup> Longhua Guo,<sup>1</sup> Jianhua Li,<sup>1</sup> and Gaolei Li<sup>1</sup>

<sup>1</sup>*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*

<sup>2</sup>*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Zhou Su; [zhousu@ieee.org](mailto:zhousu@ieee.org) and Jun Wu; [junwuhn@sjtu.edu.cn](mailto:junwuhn@sjtu.edu.cn)

Received 24 June 2016; Revised 11 September 2016; Accepted 3 October 2016

Academic Editor: Miao Wang

Copyright © 2016 Tianpeng Ye 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.

The Intelligent Transportation System (ITS) becomes an important component of the smart city toward safer roads, better traffic control, and on-demand service by utilizing and processing the information collected from sensors of vehicles and road side infrastructure. In ITS, Vehicular Cloud Computing (VCC) is a novel technology balancing the requirement of complex services and the limited capability of on-board computers. However, the behaviors of the vehicles in VCC are dynamic, random, and complex. Thus, one of the key safety issues is the frequent disconnections between the vehicle and the Vehicular Cloud (VC) when this vehicle is computing for a service. More important, the connection fault will disturb seriously the normal services of VCC and impact the safety works of the transportation. In this paper, a safety resource allocation mechanism is proposed against connection fault in VCC by using a modified workflow with prediction capability. We firstly propose the probability model for the vehicle movement which satisfies the high dynamics and real-time requirements of VCC. And then we propose a Prediction-based Reliability Maximization Algorithm (PRMA) to realize the safety resource allocation for VCC. The evaluation shows that our mechanism can improve the reliability and guarantee the real-time performance of the VCC.

## 1. Introduction

Smart city has become a critical way for sustainable urban development [1], which is composed of the smart sensor networks, Internet of Things, social network, and so forth [2–5]. The Intelligent Transportation System (ITS) is instrumental for making the road traffic in smart city safer, more efficient, and more convenient [6]. As the core component of ITS, Vehicular Ad Hoc Network (VANET) can improve the efficiency of the information exchange in road traffic. VANET is a network that vehicles can share the on-road information such as the car crash or the traffic jam on the road ahead. There are two types of communicating mechanism for vehicles to exchange their information in VANET: vehicle-to-vehicle (V2V) where vehicles communicate via on-board radio device directly and vehicle-to-infrastructure (V2I) where vehicles communicate with each other by existing infrastructure on road side. The equipment beside the road is called RSUs (Road Side Units) [7]. The ITS plays an important

role in the area of road safety with utilizing the transportation data collected by VANET. Based on these transportation data, more and more complex applications have been implemented into the vehicles to complement the ITS. In this process, the “brain” of vehicle becomes more and more “smart” as the result of the force of applications’ requirements. Some vehicles have sufficient computation and storage resources and these resources are in idle in many conditions. On the contrary, the on-board computers of some vehicles have too limited computing or storage resources to execute the service. In order to solve the imbalance of computational capability in time and space, Vehicular Cloud Computing (VCC) is proposed on the basis of VANET in last several years [8, 9]. As we all know, cloud computing is a technology that provides flexibility to the provision of computing resources by virtualizing all the distributed computing devices as a resource pool in a data center [10]. As further exploitation, VCC coordinates the computing, communication, sensing, and storage resources of the vehicles on the road to balance

the requirement of services and the limitation of hardware. In other words, aforementioned resources of the vehicles constitute the cloud resources pool.

According to the difference between the functions of the resources, services of VCC can be divided into four types, that is, "Computation-as-a-Service," "Network-as-a-Service," "Storage-as-a-Service," and "Sensing-as-a-Service" [11]. The most important difference between VCC and traditional cloud computing is that the dynamic of resource pool in VCC depends on the mobility of the vehicles. The mobility of vehicles decreases the reliability and availability of the VCC. As a matter of fact, the movement behaviors of vehicles are uncertain. In other words, a vehicle may connect or disconnect the vehicular cloud dynamically based on the complex transportation environments. A service provided to the vehicle can be decomposed into a series of tasks. Some of them required the execution results of other tasks to begin their own tasks. Therefore, there is an execution order in these tasks which is called as task sequence. If a task has been allocated to a vehicle in the Vehicular Cloud (VC), the task will be failure because of the connection fault between this vehicle and the VC. Moreover, the further task sequence which depends on failure task will be interrupted, and related services in ITS will be stopped. It is opposite with the initial goal of using VCC in ITS, which is making the road traffic safer. In traditional cloud computing systems, a lot of fault-tolerance approaches have been proposed to deal with the common node with random failures, such as the work in [12]. The authors established a real-time workflow fault-tolerant model for cloud computing and proposed a fault-tolerant scheduling algorithm which uses primary-backup model. However, these schemes cannot be applied in the vehicles of the traffic system, which have portable predictability behaviors and require high real-time services. Most of them use the backup scheme and recompute the tasks if they cause fault. The computation will introduce additional time overhead and lead to the time delay of the service execution. Thus, these existing works cannot be used directly for the VCC services with real-time requirements. One of the key safety issues in the VCC is the frequent connection faults between the vehicle and the VC when this vehicle is executing the cloud services of ITS.

To address above challenges, this paper proposes a safety resource allocation mechanism against connection fault in VCC by using a modified workflow with prediction capability. The proposed mechanism includes two phases. First, the connection fault probability model of the moving vehicles in VC is proposed. Second, a safety resource allocation algorithm, PRMA, is proposed according to the connection fault probability of the vehicle. Based on the outputs of the connection fault probability model, the safety resource allocation algorithm is to maximize the reliability of the VCC under the constraint of the service execution deadline.

The rest of this paper is organized as follows. Section 2 presents the background knowledge and related literature survey about the topic. Section 3 provides the models of the VCC and the probability models of vehicle connection fault. The formulations of connection fault in VCC are described in Section 4. Section 5 presents the safety resources allocation

workflow schedule for VCC. In Section 6, the simulations are implemented to demonstrate the efficiency of the proposed mechanisms. The conclusion and future work are shown in Section 7 at last.

## 2. Related Works

The concept of VCC has been proposed for several years. Eltoweissy et al. and Olariu et al. introduced the concept of the VC that leveraged the spare resource of the vehicles [8, 9]. In their papers, the scenario is that the vehicles can access a wireless network for long times; that is, the vehicles are parked or stuck in a traffic jam and move slowly. The work in [13] solved the problem that the low density of vehicles in some roads cannot be sufficient for the physical resources requirement of ITS by utilizing parked cars as relay nodes. In addition, the work in [14] proposed the Parked Vehicle Assistance (PVA) to make parked vehicles join the VANET. After that, Hussain et al. [15] first proposed the VENAT cloud architecture which was classified into three different architectural frameworks, VC, vehicles using cloud (VuC), and hybrid vehicular clouds (HVC).

With the steady basis of the theory system, many applications and optimization of the VCC are proposed. The first direction of the research is that focusing on the security and privacy of the VCC. Yan et al. [16] analyzed the security and privacy challenges of VCC. It is the first paper that pays attention to dealing with secure problems in VCC. After that, some security threats were concluded by several papers [17–19]. Thus this direction is a popular research point.

The second one is the efficiency of resources allocation in the VCC. The resource allocation is a traditional issue in the cloud [20]. It can improve the capability and efficiency of the VCC by optimizing the allocation strategy. All of them can be divided into several phases. The basic one is toward the VCC combining with the traditional cloud. The computing resources are in the traditional data center to be provided to the vehicles. A game-theoretical resource allocation approach was presented for central cloud resources [21]. The further one is the cloud composed of on-board computers of the vehicles. Zheng et al. proposed an SMDP-based resource allocation in VCC to maximize the total long-term reward [22].

The last one is researches on the dependability and reliability of computing resources in VCC systems. The unpredictable availability of the computational resource of the distributed vehicles has been studied in [23–26]. However, the emphasis of these researches is the variation of the vehicles in the parking lot and ignored the application scenario that the mobility of the vehicles leads to more connection faults and safety risks in VCC.

## 3. Basic Models

*3.1. The Architecture of VCC.* Generally, there are two difference architectures to construct the VCC. Although some new content delivery mechanisms were proposed [27, 28], the typical V2V and V2I communication mechanisms which are the most popular in the realistic traffic environment are considered in our paper. As shown in Figure 1, the underlayer

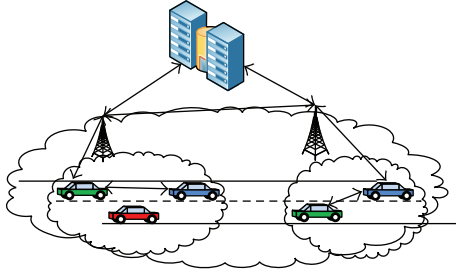


FIGURE 1: The architecture of VCC.

components are vehicles on the road. The vehicles share their resources such as computing capability, communicating bandwidth, and storage space with each other by the direct V2V communication. In our paper, this orientation model of cloud is called the V2V cloud. The advantage of this architecture is the low delay time of communication. However, due to the limited communication range of the vehicular radio, the scale of the cloud is often too small to complete some complex services such as managing evacuation when the density of vehicles is not sufficient in the range of the vehicular radio.

In order to address the limitation of this basic architecture, the RSUs are used to provide long-distance transmission for vehicles in different places. It integrates more resources in a larger area despite leading to a little bit more time delay. The most obvious disadvantage of this architecture is that it needs a huge amount of investment to build the infrastructure of this system. This kind of cloud organization model is called the V2I cloud. On the basis of these two main architectures, the VC combined with the traditional cloud becomes a Hybrid Vehicular Cloud. The traditional cloud can provide extensive storage and computing capability. This architecture can be implemented over a large geographical area where there is a data center.

Therefore, the VC should be organized by V2V if there is no communication infrastructure for the vehicles on the road and the services in demand need low latency. The VC should be organized by V2I when the following conditions are satisfied. Firstly, the communication infrastructure should be existing in this area. And then, if the large computational resources are required for the services which are provided to the vehicles, it needs to unite all the resources of the vehicles in one area.

**3.2. Integrated Architecture of Workflow and VCC.** The workflow technology has been used in traditional cloud to improve the efficiency of resources allocation [29]. A job or a service can be decomposed into a series of tasks according to the workflow theory. The dependence between them is usually described by Directed Acyclic Graph (DAG). The workflow provides some scheduling algorithms to optimize the task execution sequence or processor allocation mechanism to reduce the execution time or execution cost of the job. Naturally, the workflow can be used in VCC to improve its performance. The operation process of workflow in the VCC is shown as illustrated in Figure 2 by us. The architecture of workflow integrated with the VC can be separated into

four layers: users, virtualized resources, workflow, and physical resources. The physical resources are composed of the computation, communication, storage, and sensor resources which existed as the CPU devices, bandwidth resources, and disks. All these devices are distributed on the vehicles and RSUs. Above the physical resources layer, the workflow layer works for assigning the tasks to the most suitable physical resources to achieve an optimized execution cost. In our paper, it gets the information of physical layer by monitor module. And then, it has a specific module which is used to calculate the connection fault probability of each vehicle at a given moment. The calculation is based on the monitoring data collected from the vehicles and the environment at that specific moment. The calculation result is delivered to the workflow engine which is used to decide the resources allocation schedule. All tasks composing the job are assigned to the physical resources layer. This process integrates the distributed physical resources to be the virtualized resources. Toward users, they can use the applications in ITS to obtain the services. All the resources in demand can be provided by the virtualized resources layer through the APIs.

**3.3. Connection Fault Models in VCC.** In the last section, the literature has shown that the variation and nonreliability of vehicle nodes in the parking lot have been researched. But those papers ignored the more complex scenario where the vehicles are driving on the road. The mobility of the vehicles will lead to the connection fault between the vehicle and the VC. In order to deal with this issue, we propose a prediction model of the connection fault for the vehicles driving on the road. It contains three concrete scenarios which present the missing connection of vehicle resources. Particularly, if there is a backbone data center, all the resources can be provided by it and avoid the risk of missing connection with the vehicle resource. However, it also loses the advantage of utilizing the spare resources of the on-board computers. So, in our paper, there is no data center as the cloud resources provider in the VCC and all the scenarios are based on this assumption.

In addition, we assume that the velocity of the vehicles is a constant only if the vehicle stops itself to simplify the movement model of vehicles on the road.

**3.3.1. Scenario 1: Vehicle Leaves V2V Cloud.** The first situation of resources node missing is shown in Figure 3. The resources node aforementioned can be regarded as the vehicles which provide the resources like computing, communication, storage, and sensing resources to the VC. Three vehicles composed a mini Vehicular Cloud while they are moving on the road. When the red vehicle reaches its destination such as home, the driver of this vehicle would stop and sputter the engine very quickly for saving gasoline or electric power. Due to the shutdown of the on-board computer, this vehicle as a resource node will leave the original cloud resource pool suddenly and unpredictably. The connection fault with this vehicle leads to the task fault on this vehicle because this vehicle has no time to transport the execution result to the VC. There is no mechanism or constraint to make the on-board computer guarantee the execution result to be returned to the VC before it closed.

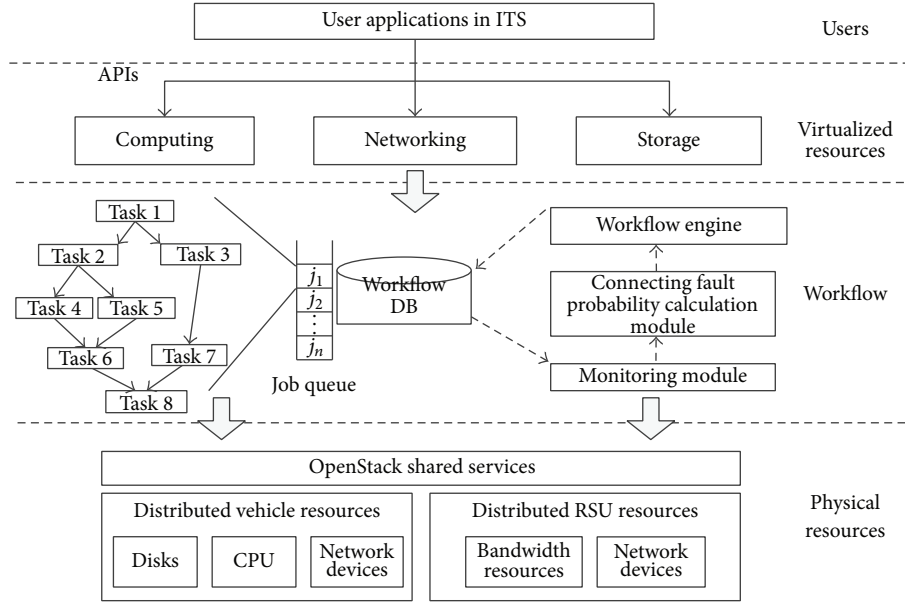


FIGURE 2: The architecture of workflow integrated with the VC.

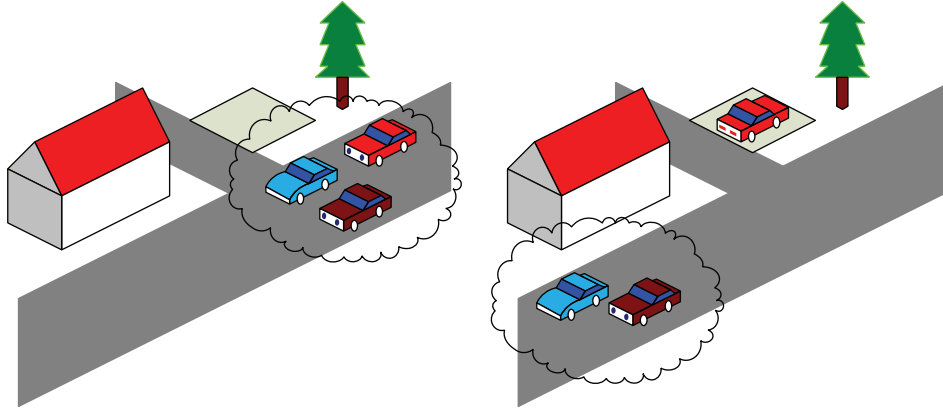


FIGURE 3: The situation of vehicle stop.

The second situation is a comprehensive set of many situations which can be called the vehicle leaving the communication range of the vehicle cluster. One of the situations can be commonly seen at the crisscross which is shown in Figure 4. There are many crisscrosses in traffic road net, especially in the city. Each vehicle would have different destination and travel path. There is a high probability that one of the vehicles constructing the VC leaves the cluster at the crisscross of the road. Even though there are some new vehicles that will join the cluster after passing the crisscross, the task processed on the leaving vehicle is failed.

The third situation is due to the difference between the velocities of the vehicles. In the realistic environment, the velocity of the vehicle is different. The vehicle chases up a vehicle cluster and joins it. However, it leaves the cluster after few minutes because of its higher speed which is shown in Figure 5. Otherwise, this vehicle is slower than the vehicle

cluster and is chased up by the cluster. Both of them can lead to the missing connection with the resource.

**3.3.2. Scenario 2: Vehicle Leaves V2I Cloud.** Scenario 2 is little different from the first one in the architecture of VC. The architecture of the first one used the V2V communication and small amount of vehicles composed the local VC. The last scenario uses the RSUs to transmit the information of all the vehicles in a large geographical area, which is shown in Figure 6, and we call it the V2I cloud. All of the authenticated vehicles share the cloud computing resource in this area. The disadvantage is that the infrastructure which can provide the communication functions must be built in advance. But the RSUs cannot spread all over the area where the vehicles want to move under the cost restriction. Hence, in the border of the VC with the RSUs, the vehicle has a high probability to migrate out to the area where there is no signal of RSUs. The

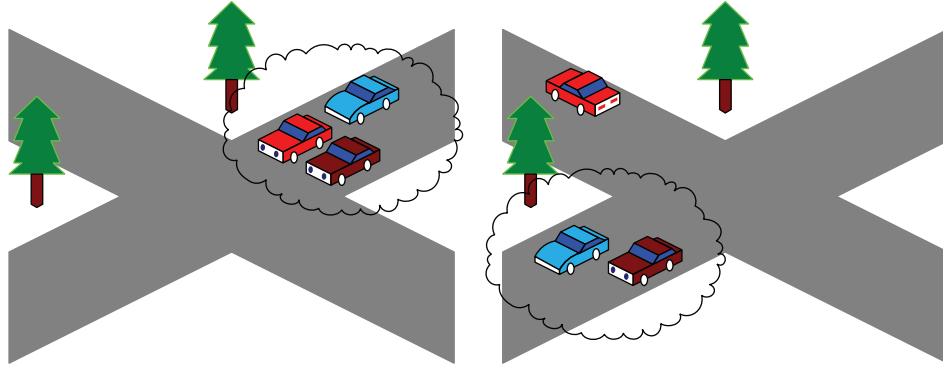


FIGURE 4: The situation of vehicle leaving the cloud at the crisscross.

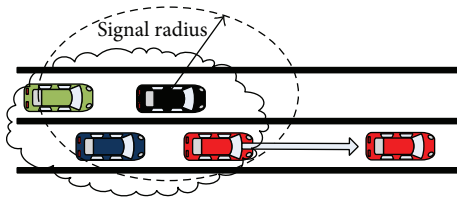


FIGURE 5: The situation of the vehicle leaving the cluster by speeding up.

resource of the vehicle will be lost if the vehicle drives out of the RSU enabled by VC in this area. This scenario is a little bit like Scenario 1, because in both of them the vehicle moves out of the signal range. But in Scenario 1, the source of the signal, the vehicle cloud cluster, is mobile. In contrary, the locations of the RSUs are stable.

The above two scenarios are very common, which leads to the missing connection of vehicle resources with the VC when the vehicle is driving on the road. In this paper, we first pay attention to predicting the probability of the occurrences of above scenarios by mathematical model. The resources with low connection fault probability are regarded as reliable. And then, we propose a workflow resource allocation algorithm to allocate the reliable resources to execute the tasks. It can improve the safety of the whole VCC system.

#### 4. Formulations of Connection Fault of VCC

**4.1. Formulation Definitions of VCC.** In this section, the clear definitions for all key concepts in the VCC will be provided and all these concepts compose the three models of the vehicle resource missing connection in the road.

**Definition 1.** The final aim of the VCC is providing services to the users driving on the road. The service for the vehicles can be substituted with job or workflow in the statement. One job can be divided into amount of tasks and the task set has a processing sequence to complete the whole services. The dependent tasks can be described by a Directed Acyclic Graph (DAG).

A DAG is defined as a 2-tuple  $G = \{\text{TASK}, E\}$ , where  $\text{TASK} = \{\text{task}_1, \text{task}_2, \dots, \text{task}_n\}$  is the set of the tasks which composed the job and  $E = \{e(\text{task}_i, \text{task}_j) \mid \text{task}_i, \text{task}_j \in T\}$

is the set of directed edges that represented the dependencies between the tasks:  $\text{task}_i$  and  $\text{task}_j$ .  $e(\text{task}_i, \text{task}_j)$  can be simplified as  $e_{ij}$  to indicate that the input data of the task  $\text{task}_j$  contains the output data of the task  $\text{task}_i$ . Hence, the task  $\text{task}_j$  cannot be executed until the output of the task  $\text{task}_i$  has been produced and transferred to the resource node running the task  $\text{task}_j$ .

**Definition 2.** The elementary unit which was allocated by service workflow schedule can be defined as task. As a minimum independent unit which can be processed in the on-board computer, it needs an input to begin the task execution and produces an output after the execution of this task. The execution time is an important factor as the basis of the service execution time. So the task can be modeled as a triple  $\text{task} = (D, w_l)$ , where  $D$  is a square matrix which contains the data length of each task, the element  $d_{i,j}$  indicates the data length needed to transfer from task  $\text{task}_i$  to task  $\text{task}_j$ , and  $w_l$  is the execution workload of this task which can be measured by Million Instructions (MI).

**Definition 3.** The vehicle is the main body of the computing resource which composes the VC, which is like the servers in the traditional cloud system. However, the vehicle has some new features like mobility, limitation of communication range, and privatization. The vehicle can be modeled as  $\text{Veh} = (C_{\text{veh}}, T_V, V, d, \text{Di}_U)$ , where  $C_{\text{veh}}$  is the computation capability of the computing resource of the on-board computer in this vehicle. The computation capability can be measured by Million Instructions Per Second (MIPS).  $T_V$  is the transmission rate (in Kbps) in the V2V mode,  $V$  is the velocity of this vehicle which is a constant assumed in advance,  $d$  is the traveling distance of this vehicle in this time which can be measured by kilometers, and  $\text{Di}_U$  is the distance away from the signal boundary of the Roadside Unit  $U$ .

**Definition 4.** The Roadside Units (RSUs) are the infrastructure of the VCC to transmit the messages in a long distance. The key factor of the RSUs is the communication range and the communication rate. It is modeled as  $U = (R_U, T_{U\text{up}}, T_{U\text{down}})$ , where  $R_U$  indicates the communication radius of the radio (in KMs) and  $T_{U\text{up}}$  and  $T_{U\text{down}}$  are the rate when the unit forwards the data to the vehicles.



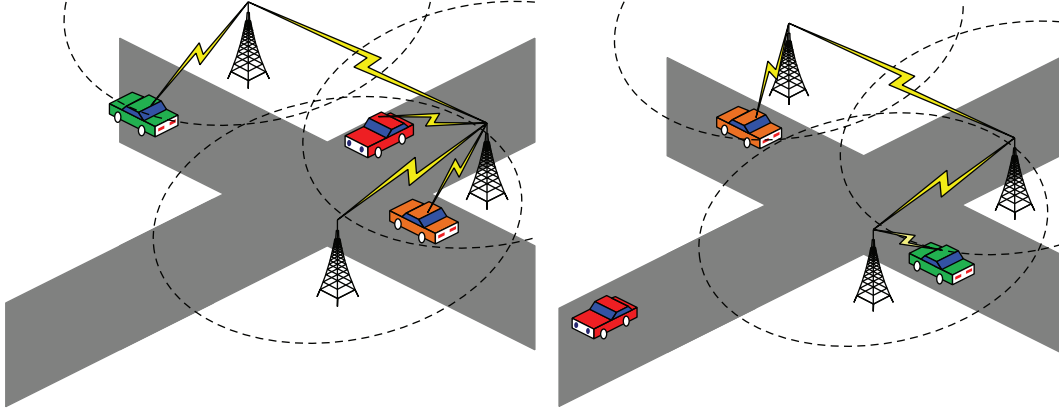


FIGURE 6: The scenario of the vehicle migrating out to the RSUs.

The scenarios of the vehicle leaving the VC have been represented in the last section. In order to avoid that the vehicle misses connection with the VC during execution of the task, a prediction-based workflow schedule mechanism is proposed. Before the workflow engine assigns the tasks to the vehicles, the probability of the vehicle missing connection with the VC will be calculated. The workflow will optimize the task allocation according to the computing result. Hence, the definition of the probability in three difference situations should be formulated. All of them are in what follows.

*Definition 5.* The total probability is composed of the probabilities of two situations which belong to Scenario 1. So total probability of the missing connection with the resource can be calculated in two different situations. The first one is in the V2V scenario:

$$P = 1 - (1 - P_s)(1 - P_l) = P_s + P_l - P_s P_l, \quad (1)$$

where  $P_s$  is the probability of the vehicle arriving at its destination during the execution time of a task and  $P_l$  is the probability of the vehicle leaving the vehicle cluster on the road.

The second one is in the V2I scenario:

$$P = P_m. \quad (2)$$

$P_m$  is the probability of the vehicle migration out of the area which has the radio signal of the RSUs. In the V2I scenario, vehicles communicate with each other via RSUs. All the vehicles in the region covered by the infrastructures are the components of the Vehicular Cloud. There are no Scenarios 1 and 2 existing, so the total connection fault probability is equal to  $P_m$ .

**4.2. Formulations for Connection Fault Models.** In this section, we will establish the functions of the probabilities in the previous proposed scenarios.

In Scenario 1, it is common sense that the destination of a vehicle can be represented as the traveling distance of the vehicle from the starting. In other words, the settlement of the stopping probability is to build the model of the traveling

distance of a vehicle. We can deduce the distribution of the traveling distance by the probability theory. According to the Stochastic Theory, the distance from the starting to the destination is a random value. We assume that the probability of the vehicle stopping in the following distance  $\Delta r$  is proportional to the distance  $r$  the vehicle has traveled. The formula is as follows:

$$P\{r < X \leq r + \Delta r \mid X > r\} = \lambda_1 \times r \times \Delta r + O(\Delta r), \quad (3)$$

where  $\lambda_1$  is a constant which has no relationship with the  $r$  and  $\Delta r$  and  $O(\Delta r)$  is the high degree polynomial of the factor  $\Delta r$ .

The left side of (3) is a conditional probability which can be transformed as follows:

$$P\{r < X \leq r + \Delta r \mid X > r\} = \frac{F(r + \Delta r) - F(r)}{1 - F(r)} \quad (4)$$

$$F(r) = \lim_{\Delta r \rightarrow 0} \frac{F(r + \Delta r) - F(r)}{\Delta r} = \lambda_1 \times r \times [1 - F(r)].$$

By solving the linear differential equation with the initial condition,  $F(0) = 0$ , we can figure out that the function of the traveling distance distribution obeys the following equations which is Rayleigh distribution. Its traveling distance cumulative probability function and probability density function are as follows:

$$F(r) = 1 - \exp(-0.5\lambda_1 r^2) \quad (5)$$

$$f(r) = \lambda_1 r \exp(-0.5\lambda_1 r^2). \quad (6)$$

The basic forms of the Rayleigh cumulative probability function and the Rayleigh density function are shown in Figure 7 and this assumption is also used in [27] which can prove that our assumption is reasonable.

In (5) and (6), the parameter  $r$  has been defined in Definition 3 as symbol  $d$  and the parameter  $\Delta r$  can be figured out by the value of the vehicle velocity and the total execution time of a task. We can calculate the total execution time of

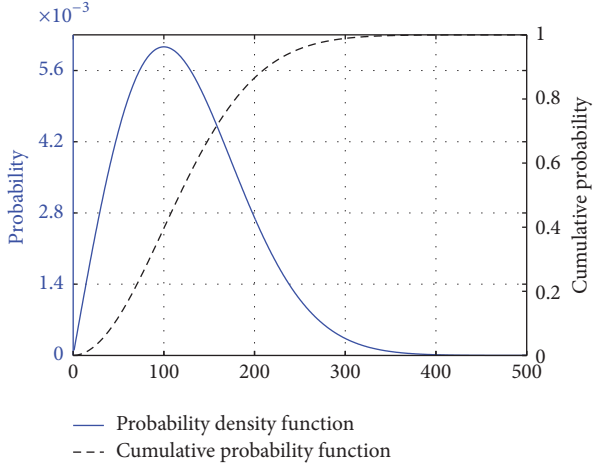


FIGURE 7: The function of Rayleigh distribution cumulative probability function and the probability density function.

the task  $task_i$  that was allocated to a vehicle defined in  $Veh = (C_{veh}, T_V, V, d, Di_U)$  by the following equations:

$$w_t = \frac{w_l}{C_{veh}} \quad (7)$$

$$c = L + \frac{d_o}{T_V} \quad (8)$$

$$Et^i = c + w_t = L + \frac{d_o}{T_V} + \frac{w_l}{C}, \quad (9)$$

where  $w_t$  is the execution time cost and  $w_l$  is the workload of a task and  $C_{veh}$  is the computation capability of the vehicle, which is defined in Definition 3.  $c$  is the time cost of the task transmission, and  $L$  indicates the communication startup cost.  $Et^i$  is the total execution time of the task  $task_i$  and the connection fault could happen in this duration.

Therefore, the parameter  $\Delta r$  can be represented as follows:

$$\Delta r = V \times Et^i. \quad (10)$$

Based on the calculated aforementioned component, the probability function (5) can be represented as follows:

$$P_s = 1 - \exp\left(-0.5\lambda_1 \left((d + V \times Et^i)^2 - d^2\right)\right). \quad (11)$$

In the second and third situations of Scenario 1, the probability distribution model can imitate from the probability model of other vehicle transportation models. A vehicle distance distribution probability model is found in [30]. It is used to analyze the vehicle collision probability. The vehicle collision event is regarded as the random event. The exponential distribution is used to describe the intervehicle space. Thus, we use random event to describe the events of the second and third situations in Scenario 1. In probability theory, the exponential distribution is used to describe the happening of the event which has no memorability. This character can be represented as

$$P(T > t + s | T > t) = P(T > s), \quad s > 0, t > 0, \quad (12)$$

which means the probability of the event happening during the time  $t$  to the time  $t + s$  is equal to the probability of the event happening during the time 0 to the time  $s$ . The equations of the exponential distribution are as follows:

$$f(x) = \lambda_2 e^{-\lambda_2 x}, \quad x > 0 \quad (13)$$

$$F(x) = 1 - e^{-\lambda_2 x}, \quad x > 0.$$

So in this formulas group, the probability of the vehicle leaving the cluster is just related with the total execution time of the task. Hence  $P_l$  can be represented as the following equation:

$$\begin{aligned} P_l &= P(t < T < t + \Delta t | T > t) = P(T < \Delta t) \\ &= 1 - e^{-\lambda_2 \Delta t}, \end{aligned} \quad (14)$$

where  $\Delta t$  is equal to  $Et^i$ , which is the whole time cost of the task  $task_i$  and  $\lambda_2$  is a constant which has no relationship with  $\Delta t$ .

In Scenario 2, due to the RSUs being stable, the probability of vehicle migration out of the RSUs region is easily described. The probability is in direct proportion to the velocity of the vehicle and the total execution time of the allocated task, in inverse proportion to the distance away from the signal boundary of the RSUs. So,  $P_m$  is as follows:

$$P_m = \lambda_3 \times \frac{V \times Et^i}{Di_U}, \quad (15)$$

where  $\lambda_3$  is a constant which relates with the realistic traffic environment. The transmission rate  $Di_U$  in this equation is determined by RSU  $U$ .

Statistically, the vehicle resource missing connection with the VC is one of main causes of the service fail. In our paper, we just discuss the service fail probability which is led by the vehicle disconnection. So the total service fail probability is just related with the probability of each vehicle allocated the tasks to execute missing connection with the VC. The event of vehicle connection fault happens independently and arbitrary one task of all fail will lead to the service fail. Hence the total service fail probability is the product of each vehicle reliability which is 1 minus the vehicle connection fault probability. It can be formulated as follows:

$$P_{total} = 1 - \prod_{j=1}^N (1 - P_j), \quad (16)$$

where  $P_j$  should be defined, respectively, in different scenarios. It is defined by (1) in V2V scenario. It is defined by (2) in V2I scenario. It is a conditional probability. The symbol  $j$  indicates the vehicle's serial number and  $N$  is the amount of total vehicles that are allocated the tasks in this VC at that moment.

The model of vehicle connection fault has been shown in the last section. By analyzing the function of the parameters  $P_s$ ,  $P_l$ , and  $P_m$ , it is obvious that the main independent variation in these functions is  $Et^i$  which decides the function

result. The probability of the vehicle missing connection will become higher if the execution time of the task running on this vehicle becomes longer. However, the connection fault probability minimum problem cannot be transformed as a problem to minimize the total execution time cost. Because each vehicle has an individual connection fault probability function, the assigning destination of the tasks also has effect on  $P_{\text{total}}$ . Therefore, a safety resources allocation schedule should be studied to minimize  $P_{\text{total}}$ .

## 5. Safety Resources Allocation Workflow Schedule for VCC

**5.1. The Fundamental Workflow Schedule.** We can firstly assume that there are  $M$  tasks that need to be allocated and  $N$  vehicles prepare to accept task at that moment. To indicate the allocation schedule, let the matrix  $X$  be an  $M \times N$  binary matrix corresponding to the schedule of  $M$  tasks into the  $N$  vehicle on-board computers. In this matrix, the element  $X_{ij} = 1$ , if the task  $\text{task}_i$  is assigned to the vehicle  $\text{Veh}_j$  and  $X_{ij} = 0$  otherwise.

$\text{Et}_{\text{total}}^p$  represents the actual execution time for multitasks on the particular vehicle  $\text{Veh}_p$ .  $\text{Et}_{\text{total}}^p$  is different from the summation of  $\text{Et}_{\text{total}}$  due to the fact that tasks which have dependence with each other could be allocated to the same vehicle. Therefore, the execution result of the parent task is unnecessary to be transferred to other vehicles. It reduces the total execution time  $\text{Et}_{\text{total}}^p$  of the whole service.

$$\text{Et}_{\text{total}}^p \leq \sum \text{Et}^i. \quad (17)$$

We can define  $\text{Et}_{\text{total}}^p$  as follows: Firstly, the relationship of the tasks assigned to the vehicle  $\text{Veh}_p$  can be shown in the matrix  $X$ . The  $p$ th row in  $X$  shows that the task  $\text{task}_i$  is assigned to this vehicle when  $x_{ip} = 1$ . we define a new matrix  $Y^p$  to represent the possible dependence between the tasks allocated to vehicle  $\text{Veh}_p$ :

$$\begin{aligned} y_{i_1 i_2} &= x_{i_1 p} x_{i_2 p} \quad i_1 \neq i_2 \\ y_{i_1 i_2} &= 0 \quad i_1 = i_2, \end{aligned} \quad (18)$$

where  $1 \leq i_1, i_2 \leq M$ , and  $y_{i_1 i_2}$  is the element in the matrix  $Y^p$ .

The actual dependence between the tasks is shown in the set  $E = \{e(\text{task}_i, \text{task}_j) \mid \text{task}_i, \text{task}_j \in \text{TASK}\}$ , and it can be transferred to a matrix  $E$ , and the element  $e_{ij} = 1$  when  $e(\text{task}_i, \text{task}_j)$  existed in the set  $E$ .

So, the actual existing dependence between the tasks assigned to the vehicle  $\text{Veh}_p$  which is named as  $E_{\text{act}}^p$  and the element in it can be indicated as  $e_{ij}^{\text{act}}$ :

$$E_{\text{act}}^p = Y^p \cdot E. \quad (19)$$

So,  $\text{Et}_{\text{total}}^p$  can be calculated by the following equation:

$$\text{Et}_{\text{total}}^p = \sum_{i=1}^M x_{ip} \text{Et}^i - \sum_i \sum_j e_{ij}^{\text{act}} \left( \frac{d_o^i}{T_{\text{pup}}} + \frac{d_i^j}{T_{\text{pdown}}} \right). \quad (20)$$

**5.2. Priority Classification of Tasks.** After the derivation of  $\text{Et}_{\text{total}}^p$ , to design a workflow schedule, we need to determine the priority compartment for each task to confirm the execution order of these tasks. In this paper, we determine these parameters according to the dependence of each task in the workflow. Then, due to the deadline of the service, we have to separate it into each task to find out the Earliest Starting Time (EST), Earliest Finishing Time (EFT), Latest Finishing Time (LFT), Actual Starting Time (AST), and Actual Finishing Time (AFT) of each task. At last, we can find out the allocation strategy under the task priority and time constraint.

At the beginning, we have to consider some definition used in the following priority computation. In (7),  $w_i$  refers to the computing time cost of a task which is executed on a vehicle. Therefore, this result can be extended to all the combinations of the tasks and vehicles. A matrix  $W_{M \times N}$  is set to store the computing time cost, where  $M$  is the number of tasks and  $N$  is the number of the vehicles. The element  $w_t^{i,j}$  in the matrix refers to computing time cost of task <sub>$i$</sub>  assigning to Veh <sub>$j$</sub> . So, we can get the average execution time for each task:

$$\bar{w}_t^i = \frac{\sum_{j=1}^N w_t^{i,j}}{N}. \quad (21)$$

The average communication cost for the edge  $e = \{(\text{task}_i, \text{task}_j) \mid \text{task}_i, \text{task}_j \in \text{TASK}\}$  can be inferred from (8):

$$\bar{c}_{i,j} = \bar{L} + \frac{d_{i,j}}{T_V}, \quad (22)$$

where  $\bar{T}_V$  is the average transfer rate among the vehicles in the VC and  $\bar{L}$  is the average communication startup time cost.

According to [31], the task can be classified into simple task and synchronization task. The task has more than one parent task or its child task is the synchronization task. Otherwise, this task is the simple task. The simple tasks executed between two synchronization tasks are called branches. We can set the same deadline for the several branches between two synchronization tasks.

According to [32], the upward rank of a task  $\text{task}_i$  is recursively defined as

$$\begin{aligned} \text{rank}(\text{task}_i) &= \bar{w}_t^i \\ &+ \max_{\text{task}_j \in \text{succ}(\text{task}_i)} (\bar{c}_{i,j} + \text{rank}(\text{task}_j)), \end{aligned} \quad (23)$$

where  $\text{succ}(\text{task}_i)$  is the set of immediate successors of task  $\text{task}_i$ ,  $\bar{c}_{i,j}$  is the average communication time cost of edge  $e(\text{task}_i, \text{task}_j)$ , and  $\bar{w}_t^i$  is the average computing time cost of task  $\text{task}_i$ . And the upward rank starts from the exit task  $\text{task}_{\text{exit}}$ . The rank value of the exit task is determined by the following equation:

$$\text{rank}(\text{task}_{\text{exit}}) = \bar{w}_t^{\text{exit}}. \quad (24)$$

So, we can transfer the DAG which is used to describe the task execution time sequence into the rank value for each task to represent its execution time sequence.



**5.3. Task Execution Deadline Schedule.** The EST, EFT, and LFT attributes need to be defined. The EST for entry task  $\text{task}_{\text{entry}}$  can be indicated as follows:

$$\text{EST}(\text{task}_{\text{entry}}, \text{Veh}_p) = 0. \quad (25)$$

Based on the EST for entry task, the EST, EFT, and LFT values for other tasks are shown in the following equations:

$$\begin{aligned} & \text{EST}(\text{task}_i, \text{Veh}_p) \\ &= \max \left\{ \begin{array}{l} \text{avail}[p], \\ \max_{\text{task}_m \in \text{pred}(\text{task}_i)} (\text{AFT}(\text{task}_m) + c_{m,i}) \end{array} \right\} \quad (26) \\ & \text{EFT}(\text{task}_i, \text{Veh}_p) = w_t^{i,p} + \text{EST}(\text{task}_i, \text{Veh}_p), \end{aligned}$$

where  $\text{avail}[p]$  is the ready time of the vehicle  $\text{Veh}_p$  to execute  $\text{task}_i$  for the situation that there has been a task assigned to the vehicle before  $\text{task}_i$  and  $\text{pred}(\text{task}_i)$  is the set of immediate predecessor tasks of the task  $\text{task}_i$ . In order to start the execution of the task  $\text{task}_i$ , all immediate predecessor tasks must have been finished. After the task  $\text{task}_m$  is assigned to the vehicle  $\text{Veh}_p$ , the EST and the EFT of the next task are set to be equal to the AST and the AFT, respectively.

**5.4. The Fitness Function.** In order to suffice the real-time requirement of a service, it must set a deadline for the total service execution time. If the computing resources of the local on-board computer are insufficient to finish the execution before the deadline, the users have to offload the tasks to the VC. It can optimize the total service execution time by using better allocation schedule. However, the first aim of our paper is to guarantee the safety of the service. So, our resource allocation is to minimize the total service fail probability under the constraint of the total execution time. It can improve the safety of the VCC by increasing the reliability of the resource during the service generation process.

$$\begin{aligned} & \text{LFT}(\text{task}_{\text{exit}}) = \text{Deadline} \\ & \text{LFT}(\text{task}_i) \\ &= \min_{\text{task}_j \in \text{succ}(\text{task}_i)} ((\text{LFT}(\text{task}_j) - w_t^{i,s} - c_{i,j,s,r})). \end{aligned} \quad (27)$$

Deadline is the deadline for the exit task  $\text{task}_{\text{exit}}$  as well as the whole workflow. According to the deadline for the whole service, the LFT can be seen as the deadline for each task. The LFT can be calculated in (27).

The aim of our schedule is to minimize the probability of service fail due to the connection fault of vehicle resources in VC and promise that all the tasks can be finished before deadline. The fitness function can be formulated as follows:

$$\begin{aligned} & \text{AFT}(\text{task}_i, \text{Veh}_p) \leq \text{LFT}(\text{task}_i), \quad \text{at } \forall i \\ & \min (P_{\text{total}}). \end{aligned} \quad (28)$$

**5.5. Prediction-Based Reliability Maximization Algorithm (PRMA).** In fact, the heuristic algorithm has been used widely to solve the optimization problems of the workflow [33, 34]. The heuristic algorithm is always used to find the locally optimal solution or suboptimal solution. The proposed safety resource allocation mechanism is also based on the workflow theory, so the heuristic algorithm can be used to solve the optimization problem in the proposed mechanism.

In this paper, we use the principle of greedy algorithm, which is one kind of heuristic algorithms. The proposed algorithm follows making the choice which has the locally minimized probability at each task assignment with the hope of finding a global minimized  $P_{\text{total}}$ . The service fail probability of each vehicle depends on the states of the vehicle and the environment at some moment. It is difficult to make a global decision because it lacks the future information of the vehicle and environment. Therefore, all the decisions should be made locally in the time domain. Thus, the proposed algorithm uses the principle of making locally optimal choice to reduce the value of  $P_{\text{total}}$  at most. In addition, to decrease the probability of the optimized result becoming the locally minimized probability, the randomized greedy approach is widely used [35, 36]. The proposed algorithm selects randomly some other vehicles besides the vehicle with minimized probability. The vehicle with the minimized probability and aforementioned random vehicles are all selected as the candidate destinations of task assigning. Then, in the vehicular communication fault probability computation, the probabilities of each vehicle under different parent vehicles will be computed, respectively. All the computed probabilities will be compared together. After that, the vehicle which leads to the temporary minimized service fail probability and some random vehicles will be selected as the candidate task assigning destinations. The above process will go on until the last task in the task sequence is assigned. The whole process can decrease the possibility of the optimized result being trapped in the local optimization by randomized algorithm. Before that, the AFT of this task has to be less than the LFT to ensure the whole service can be finished before the deadline. So, the algorithm we proposed, PRMA, is shown in Algorithm 1. In this algorithm, the number of the aforementioned random vehicles is set to two, for example. The number  $n$  of the symbols  $\mathbf{A}_n$ ,  $\mathbf{B}_n$ , and  $\mathbf{C}_n$  refers to the order number of the task in the task sequence and  $n+1$  refers to the next task that will be assigned.

## 6. Evaluations

We simulate the process of the tasks assigned to the vehicles. In the simulation, the service executed in the VCC is invariable. It means that the relationship among the tasks is confirmed before the simulation beginning. Therefore, the degrees of tasks, which are used to indicate the number of their children tasks, are constants in the simulation. However, there are also some random parameters in the simulation. Vehicles have different velocities, travel distances, communication capabilities, and computational capabilities. In addition, each task composing the service has different

- (1) Input the computation cost of tasks, communication cost of edges with average value and the Deadline of this service.
- (2) Calculate LFT of each task, starting from the exit task.
- (3) Compute  $\text{rank}_u$  for each task by DAG, starting from the exit task.
- (4) Sort the tasks in a sequence by non-increasing order of  $\text{rank}_u$  value.
- (5) **For** each task in the sequence order sorted at step (3) **do**
  - (6) **For** in the condition of each vehicle (the optimized vehicle  $A_n$  and two other random vehicles  $B_n$  and  $C_n$ ) chosen at the last loop **do**
    - (7) Calculate the AFT of this task allocated to every available vehicle in the VC.
    - (8) **If** AFT of task on one vehicle smaller than the LFT.
    - (9) Calculate  $P_{\text{total}}$  for the vehicle in the condition that the last loop chosen vehicle is  $A_n$ ,  $B_n$  or  $C_n$ .
    - (10) **End If**
  - (11) **End For**
  - (12) Choose the vehicle  $A_{n+1}$  which has the minimum probability to occur connection fault at that moment from all the vehicles as the branches of three vehicles chosen from last loop and other two random vehicles  $B_{n+1}$  and  $C_{n+1}$  as the allocation destination of this task.
- (13) **End For**

ALGORITHM 1: The PRMA.

TABLE 1: The range of the parameters values in simulation.

Parameter	Value	Unit
Vehicle number	5~30	/
Task number	30	/
Degrees for each task	0~3	/
Workload	20~70	MI
Computing capability	5~10	MIPS
Communication capability	7~13	Mbps
Average communication startup time cost	3	s
$\lambda_1$	$0.9 \times 10^{-5} \sim 1.1 \times 10^{-5}$	/
$\lambda_2$	$0 \sim 10^{-4}$	/
$\lambda_3$	0.05	/
Velocity	10~20	m/s
Communication data	1~40	Mb
Travel distance	0~100	Km
Distance to radio boundary	4000~6000	m
Service fail probability	0~1	/

random parameters, such as the amount of the communication data and workload. Due to the complexity of the real vehicular and traffic environment, we present the range of the parameters in our simulation for the algorithms as a matter of experience as in Table 1.

First, to get a better value of the deadline for workflow allocation schedule, we simulate the execution using the Heterogeneous-Earliest-Finish-Time (HEFT) Algorithm, which is the well-known makespan minimization algorithm. It is an application scheduling algorithm for a bounded number of heterogeneous processors to schedule each selected task on its “best” processor, which minimizes the task’s execution time [32]. The execution time of the service using the HEFT schedule is used as the parameter of the deadline calculation in our workflow schedule.

$$\text{Deadline} = \alpha \cdot \text{Et}_{\text{HEFT}}, \quad (29)$$

where  $\alpha$  is a factor and it is always set in the range from 1.5 to 5 [37]. We assume  $\alpha = 1.5$  in our simulation.

The average execution time of algorithm HEFT and PRMA is shown in Figure 8. It is the average value of 20 times simulations and each time uses the new random parameters. As can be seen, the execution time of PRMA is smaller than 1.5 times of the value of HEFT which is set as the deadline for the service accomplishment using PRMA. In other words, the PRMA can suffice the requirement with the constraint of deadline. On the other hand, with the increase of the number of vehicles, which means the sufficiency of computing resources, the average execution time of the whole service is decreased from 135.7 s to 118.2 s (HEFT) and from 181.5 s to 150.6 s.

The service fail probability can be calculated by (12). We also use the vehicle connection fault probability model to calculate the service fail probability of HEFT as a contrast.

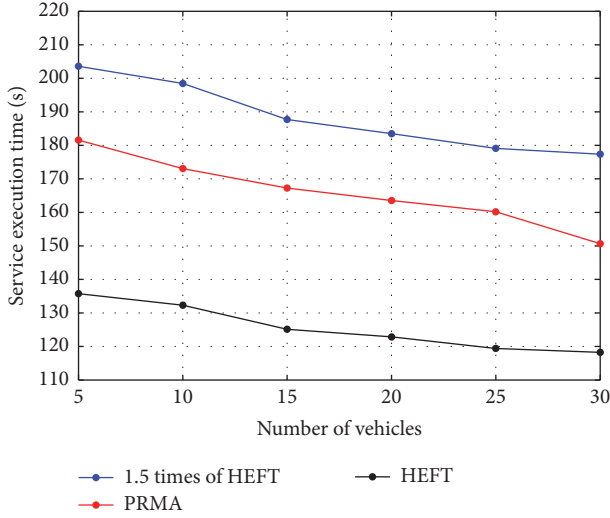


FIGURE 8: Average execution time for whole service depends on vehicle numbers.

The comparison between two algorithm service fail probabilities of V2V is shown in Figure 9. We can see that the service fail probability of PRMA is much smaller than HEFT. The probability of HEFT is about 4.9% and the probability of PRMA is about 2%. With the increase of the number of vehicles, which means the task allocation destination has more choices, the probability of PRMA shows the decreasing trend from 2.04% to 0.80%. However, the probability of HEFT does not show this trend because its resources allocation is based on makespan minimization which is a random value on the vehicle connection fault probability. In addition, the scenario of V2I is also simulated. The number of vehicles in V2I scenario is much larger than V2V. The changes of the service fail probability are less than 0.1% when the number of vehicles increases from 100 to 200. This is because the task allocation destination has enough choices and the increase of vehicle numbers has no obvious influence on the result.

Figures 10 and 11 show that the average service fail probability depends on vehicle velocity in V2I and V2V, respectively. In V2I scenario, the service fail probability of PRMA increases from about 0.45% to 21.4% when the mean value of the vehicle velocity increases from 5 m/s (18 km/h) to 30 m/s (108 km/h). And the probability of HEFT increases from about 3.7% to 35.2% in the same condition. The increase rate of HEFT is larger than the increasing rate of our scheme because the vehicles, which have a higher velocity, have a higher possibility to leave the RSUs signal area than the low-velocity vehicles during the same time.

Figure 12 shows that the average service fail probability depends on communication capability in V2I and V2V, respectively. We can see that the probabilities decrease with the increase of the communication capability mean value in these two scenarios. In V2I scenario, the service fail probability of PRMA decreases from about 13.4% to 7% when the mean value of communication capability increases from 5 Mbps to 30 Mbps. In V2V scenario, this parameter decreases from about 1.57% to 0.96%. All of them are smaller

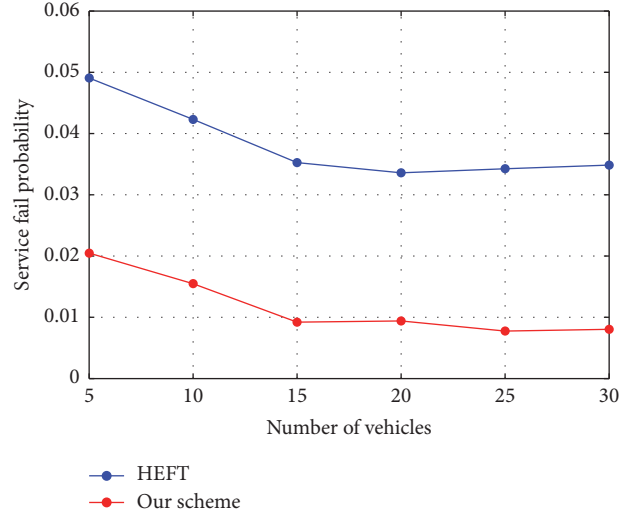


FIGURE 9: Average service fail probability depends on vehicle numbers.

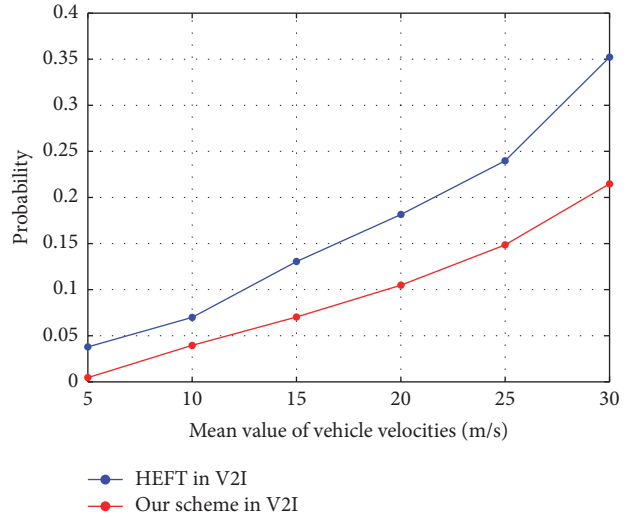


FIGURE 10: Average service fail probability depends on vehicle velocity in V2I.

than the values of HEFT. The communication time cost is one of the main components of the service total execution time. The increase of the communication capability is one of the main causes which lead to the reduction of the service execution time. Thus, according to (9), (14), and (15), the service fail probability will decrease.

In summary, we can conclude that the PRMA can increase the reliability of the VCC during the service execution duration and suffice the requirement of deadline constraint.

## 7. Conclusions

The vehicle connection fault issue will lead to the service failure of VCC in ITS, which makes the transportation unsafe. To address this challenge, a safety resource allocation mechanism against the connection fault for VCC was proposed in this paper. In this mechanism, the connection fault

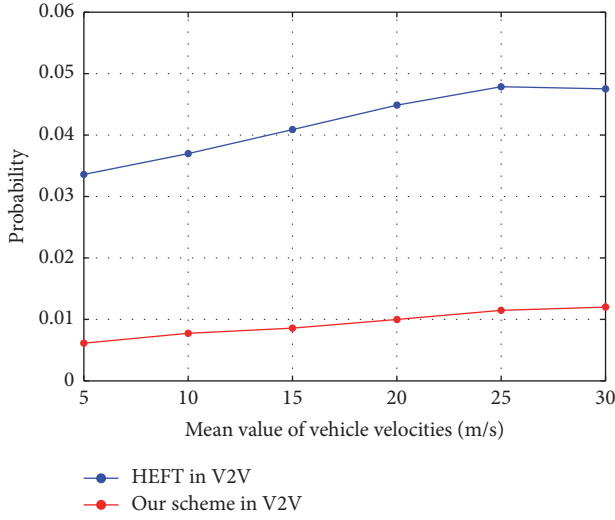


FIGURE 11: Average service fail probability depends on vehicle velocity in V2V.

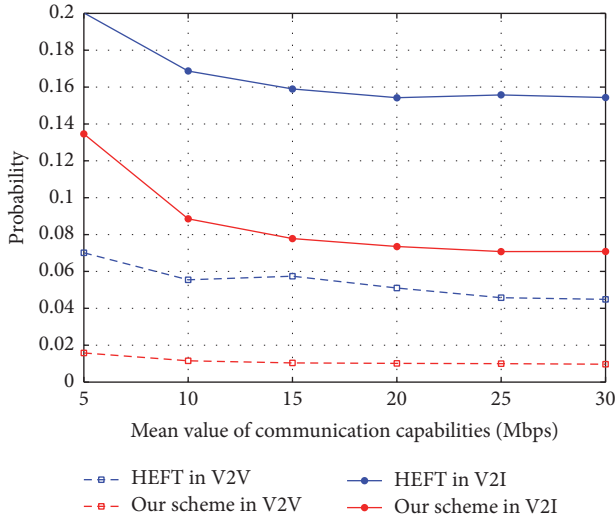


FIGURE 12: Average service fail probability depends on communication capability.

probability models of the vehicles traveling on the road were built and they were used to predict the vehicle connection fault probability according to the real-time monitoring data from the VC. After that, the prediction results can be used by the workflow schedule to improve the reliability of the task assignment. An heuristic algorithm PRMA which is based on the greedy algorithm and task priority calculation method were proposed to minimize the probability of connection fault. Thus VCC can complete the service before the deadline to suffice the real-time requirement of the services in ITS. Evaluations show that our mechanism can improve the reliability and real-time performance of VCC. The future work is to optimize the prediction model of vehicle connection fault probability. In the real traffic environment, many other factors also will lead to the communication fault, such as the radio shielding by the building and the fluctuation of the

terrain. All these factors could be considered in the future to make the prediction model more precise. The heuristic algorithm also can be optimized in the aspect of improving the balance between the computational complexity and the optimized results.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper. In addition, the authors confirm that the mentioned received funds in the "Acknowledgment" do not lead to any conflict of interests regarding the publication of this manuscript.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants nos. 61571286, 61401273, and 61431008.

## References

- [1] G. Pan, G. Qi, W. Zhang, S. J. Li, Z. Wu, and L. T. Yang, "Trace analysis and mining for smart cities: issues, methods, and applications," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 120–126, 2013.
- [2] J. Wu, K. Ota, M. Dong, and C. Li, "A hierarchical security framework for defending against sophisticated attacks on wireless sensor networks in smart cities," *IEEE Access*, vol. 4, pp. 416–424, 2016.
- [3] L. Guo, J. Wu, Z. Xia, and J. Li, "Proposed security mechanism for XMPP-based communications of ISO/IEC/IEEE 21451 sensor networks," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2577–2586, 2015.
- [4] J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, May 2010.
- [5] Z. Su, Q. Xu, and Q. Qi, "Big data in mobile social networks: a QoE-oriented framework," *IEEE Network*, vol. 30, no. 1, pp. 52–57, 2016.
- [6] G. Al-Kubati, A. Al-Dubai, L. Mackenzie, and D. P. Pezaros, "Stable infrastructure-based routing for intelligent transportation systems," in *Proceedings of the IEEE International Conference on Communications (ICC '15)*, pp. 3394–3399, London, UK, June 2015.
- [7] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network & Computer Applications*, vol. 37, no. 1, pp. 380–392, 2014.
- [8] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in *Ad Hoc Networks: Second International Conference, ADHOCNETS 2010, Victoria, BC, Canada, August 18–20, 2010, Revised Selected Papers*, vol. 49 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 1–16, Springer, Berlin, Germany, 2010.
- [9] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, no. 1, pp. 7–21, 2011.



- [10] M. Guzek, P. Bouvry, and E.-G. Talbi, "A survey of evolutionary computation for resource management of processing in cloud computing," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 53–67, 2015.
- [11] L. Gu, D. Zeng, and S. Guo, "Vehicular cloud computing: a survey," in *Proceedings of the IEEE Globecom Workshops (GC Wkshps '13)*, pp. 403–407, Atlanta, Ga, USA, December 2013.
- [12] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, "Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds," *IEEE Transactions on Parallel and Distributed Systems*, 2016.
- [13] D. Eckhoff, C. Sommer, R. German, and F. Dressler, "Cooperative awareness at low vehicle densities: how parked cars can help see through buildings," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–6, December 2011.
- [14] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *Proceedings of the IEEE INFOCOM 2011*, pp. 431–435, April 2011.
- [15] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking vehicular communications: merging VANET with cloud computing," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 606–609, Taipei, Taiwan, December 2012.
- [16] G. Yan, D. B. Rawat, and B. B. Bista, "Towards secure vehicular clouds," in *Proceedings of the 6th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS '12)*, pp. 370–375, July 2012.
- [17] M. K. Sharma and A. Kaur, "A survey on vehicular cloud computing and its security," in *Proceedings of the 1st International Conference on Next Generation Computing Technologies (NGCT '15)*, pp. 67–71, Dehradun, India, September 2015.
- [18] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, "Security challenges in vehicular cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 284–294, 2013.
- [19] F. Ahmad, M. Kazim, A. Adnane, and A. Awad, "Vehicular cloud networks: architecture, applications and security issues," in *Proceedings of the IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC '15)*, pp. 571–576, Limassol, Cyprus, December 2015.
- [20] Z. Su, Q. Xu, M. Fei, and M. Dong, "Game theoretic resource allocation in media cloud with mobile social users," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1650–1660, 2016.
- [21] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward Cloud-based vehicular networks with efficient resource management," *IEEE Network*, vol. 27, no. 5, pp. 48–55, 2013.
- [22] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An SMDP-based resource allocation in vehicular cloud computing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7920–7928, 2015.
- [23] P. Ghazizadeh, R. Mukkamala, and R. Fathi, "Modeling and predicting fault tolerance in Vehicular Cloud Computing," in *Proceedings of the International Conference on Computing and Network Communications (CoCoNet '15)*, pp. 395–400, Trivandrum, India, December 2015.
- [24] P. Ghazizadeh, R. Florin, A. G. Zadeh, and S. Olariu, "Reasoning about mean time to failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 751–761, 2015.
- [25] P. Ghazizadeh, S. Olariu, A. G. Zadeh, and S. El-Tawab, "Towards fault-tolerant job assignment in vehicular cloud," in *Proceedings of the IEEE International Conference on Services Computing (SCC '15)*, pp. 17–24, New York, NY, USA, June 2015.
- [26] R. Florin, P. Ghazizadeh, A. G. Zadeh, and S. Olariu, "Enhancing dependability through redundancy in military vehicular clouds," in *Proceedings of the IEEE Military Communications Conference (MILCOM '15)*, pp. 1064–1069, Tampa, Fla, USA, October 2015.
- [27] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: a survey and future perspectives," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 98–104, 2016.
- [28] Z. Su, Y. Hui, and S. Guo, "D2D-based content delivery with parked vehicles in vehicular social networks," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 90–95, 2016.
- [29] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317–3327, 2015.
- [30] J. Yu and R. Buyya, "Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms," *Scientific Programming*, vol. 14, no. 3–4, pp. 217–230, 2006.
- [31] C. Garcia-Costa, E. Egea-Lopez, J. B. Tomas-Gabarron, J. Garcia-Haro, and Z. J. Haas, "A stochastic model for chain collisions of vehicles equipped with vehicular communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 503–518, 2012.
- [32] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.
- [33] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 222–235, 2014.
- [34] H. M. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1203–1212, 2013.
- [35] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *Proceedings of the IEEE INFOCOM*, pp. 1293–1302, 2000.
- [36] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," in *Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS '12)*, pp. 649–658, 2012.
- [37] S. Abrishami, M. Naghibzadeh, and D. Epema, "Cost-driven scheduling of grid workflows using Partial Critical Paths," in *Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (GRID '10)*, pp. 81–88, Brussels, Belgium, October 2010.





Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

