Research of vehicular cloud storage resource management based on trust mechanism

Jin Wang, Miao Zhang, Xiang Gu, Tiequan Wang, Jie Wang and Liang Chen

Abstract
This article focuses on how to effectively make full use of the storage resources in vehicular cloud. A trust mechanism called DI-Trust (Trust Model Based on Dynamic Incentive Mechanism) is proposed to schedule vehicular cloud storage resource. The discussion is under the scenario of the parking lot where vehicle nodes are in static state. The model can reasonably arrange the suitable scheduling algorithm according to the attribute characteristics of different kind of service requirements. The trust value of a vehicle is updated according to the model to fully utilize the vehicle idle storage resources. The simulation experiment results show that the model can work effectively. It can objectively evaluate trust values of vehicle nodes, and construct the effective resource schedule of the vehicular cloud storage resource to meet needs from users.

Keywords
Vehicular cloud, trust mechanism, resource scheduling, information security

Date received: 17 February 2021; accepted: 1 March 2021

Handling Editor: Peio Lopez Iturri

Introduction
The new development in the field of VANET (Vehicle Ad hoc Network) is the integration of cloud computing and VANET, namely, the Vehicular Cloud (VC), which is similar to the system architecture of mobile cloud computing. The structure of VC1 can be defined into three layers: the vehicle interior, communication, and cloud layer. The vehicle interior layer is responsible for monitoring or collecting all kinds of information in vehicle through various sensors to send information to cloud for storage. The communication layer involves two basic communication modes of VANET: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. V2V communication takes charge of transmitting vehicle information to surrounding vehicles and clouds in a timely manner. V2I communication is able to accurately understand the traffic conditions within the communication range and improve the safety of vehicles. Cloud layer can not only meet the real-time requirements of vehicle users but also integrate information data to provide users with a variety of services, such as Network as a Service (NaaS), Storage as a Service (SaaS), and Information as a Service (IaaS). It is of great significance to excavate the potential of vehicle resources for the development of the VANET.

School of Information Science and Technology, Nantong University, Nantong, China

Corresponding author:
Xiang Gu, School of Information Science and Technology, Nantong University, Nantong 226019, China.
Email: gu.x@ntu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
The important issues affecting vehicle cloud services are the same as the VANET\textsuperscript{2} and mobile ad hoc network (MANET),\textsuperscript{3} including data security, user privacy, and access control; the customer’s main desire for a cloud is to get quality service according to his needs and security. Trust mechanism will be helpful to solve these two problems. While there are many challenges to building trust model in a VC. First, the high mobility of the vehicle makes it difficult for a vehicle to evaluate the received data in real time, and then it is difficult to confirm the credibility of a vehicle. Second, decentralized architecture and open environment of the VC make it difficult to collect sufficient information about a vehicle being evaluated and to establish a long-term stable relationship between the vehicles.\textsuperscript{5} Third, linking trust to vehicles and the data they may report is important for the VC.\textsuperscript{5} Generally, it is a challenge to add trust mechanism reasonably and effectively to ensure the security and integrity of resource scheduling in the process of inter-vehicle resource scheduling.\textsuperscript{6}

The vehicle cloud resource management strategy of trust mechanism can help the resource scheduling system to operate effectively in an open and dynamic environment. Many scholars have done a lot of research in this field. The proposed trust models are mainly based on vehicle entity-centered trust model,\textsuperscript{7} incentive model for suppressing malicious nodes,\textsuperscript{8} hybrid trust model for vehicle reliability evaluation,\textsuperscript{9} solution for virtual machine migration of vehicle cloud,\textsuperscript{10} cooperative adaptive resource scheduling method for vehicle cloud,\textsuperscript{11} alliance game model based on bilateral matching theory,\textsuperscript{12} and so on.

This article mainly studies two problems. One is the application scenario based on the parking environment, which combines the vehicle node in the parking lot with the cloud server to construct the vehicle cloud platform, and the vehicle node can communicate with the cloud server when providing the service and obtaining the service. However, a trust model is designed to guarantee the scheduling of vehicle node storage resources, and the corresponding algorithm is designed to improve the operational efficiency and network security of the vehicle cloud platform.

**Trust model based on dynamic incentive mechanism**

**Trust model based on dynamic incentive mechanism architecture**

This article mainly studies the trust model under the centralized network architecture in the vehicle cloud environment. The application scenario is a large parking lot. There is a public cloud server installed in the parking lot. As shown in Figure 1, wireless communication is used between vehicle nodes and a roadside unit (RSU), and wired communication is used between the RSU and the public cloud. RSUs are responsible for forwarding data communication between the public cloud server and vehicle nodes.

**Initial trust values of vehicle node**

When a new vehicle node enters the environment, it has no transaction record in the cloud server, and in order for it to be added to the environment to provide services, it must be given an initial value of trust. We need to choose an initial value that is appropriate for the entire scenario to avoid the adverse effects of too high or too low. In this article, the value range of the trust value is (0,1).\textsuperscript{13} When there are fewer vehicle nodes, the initial service value of the newly added vehicle node is 0.4. When the vehicle node reaches a certain number, the initial trust value of the newly added node will be given to the average value of all exists vehicle’s trust values, which can give the new node a fair chance.

**Calculation of trust values**

This article synthesizes the characteristics and the interaction scale of the service providing vehicle node to calculate and update its trust value. The calculation of a node’s trust value can be expressed as formula (1)

\[
Dtr(i) = \alpha_1 Utr(i,n) + \beta_1 Ttr(i), \quad \text{where } \alpha_1 + \beta_1 = 1 \text{ and } 0.1 < \alpha_1 < 0.3
\]

where \(Dtr(i)\) represents the total trust value of the service providing vehicle node \(i\), \(Utr(i,n)\) represents the characteristic property value of the node \(i\) in the last \(n\) days. \(Ttr(i)\) represents the interactive scale value of the node \(i\). \(\alpha_1\) and \(\beta_1\) represent the proportion of \(Utr(i,n)\) and \(Ttr(i)\) in the total trust value \(Dtr(i)\). And because the correlation between total trust value and interaction scale is greater than characteristic property, here set \(\alpha_1 < \beta_1\). In order to accurately distinguish the proportion of \(Utr(i,n)\) and \(Ttr(i)\), this article defines \(0.1 < \alpha_1 < 0.3\) and \(\beta_1 = 1 - \alpha_1\).

**Calculation of characteristic property value**

The characteristic property value is used to count the vehicle node’s recent stay in the parking lot. The longer the stay time, the more regular the parking period, the better it is. In the parking environment, each vehicle node will record its parking time in the public Cloud Server. In the time record table, a day will be divided into four time periods depending on the busyness situation: 0 to 6 o’clock for night parking time period, 6 to 12 o’clock for morning parking period, 12 to 18 o’clock for afternoon parking time period, and 18 to 24 o’clock.
for evening parking period. With the most recent \( n \) days as the cycle, the parking status of a vehicle node in these most recent \( n \) days is recorded in the parking schedule. When the public cloud server wants to select service vehicle nodes, it will calculate the parking time coverage of all vehicles in the current time period and the next time period through the parking schedule. As shown in Figure 2, several days parking time information for the vehicle node is listed. The red block indicates that the vehicle is parked in the parking lot during this time period.

The derivation of all formulas in the following and the background of the environment are assumed that the current time is 10:00, and that the other time calculation methods are consistent with the method described below.

Assuming that the current time is 10 o’clock and a vehicle node in the parking lot requests a storage resource service. Then, the public cloud server starts calculating the parking time coverage of all vehicles in the parking lot immediately after receiving that request. And as the current time is 10 o’clock, it is the morning parking period. Therefore, the public Cloud Server only needs to calculate the morning parking period and the afternoon parking time period of the parking data. The advantage of this is that the public cloud server calculates only half of the data in the parking schedule. The calculation method is expressed in formula (2)

\[
\eta(i, n) = \frac{n \sum_{j=1}^{n} w_j}{12n}, \quad \text{where} \quad 0 < \eta(i, n) < 1 
\]

In the above formula, \( \eta(i, n) \) means the parking time coverage of the vehicle node \( i \) on the nearest \( n \)-days, \( w_j \) indicates the time length of the vehicle node \( i \) stopped in the morning and afternoon period in day \( j \), \( \sum_{j=1}^{n} w_j \) represents the total time of the vehicle node \( i \) stopping in the morning and afternoon periods on those \( n \) days. The higher the \( \eta(i, n) \) is, the higher the probability that the vehicle node \( i \) will exist in the morning and afternoon periods. Accordingly, the longer the vehicle node \( i \) can provide storage resources in the morning and afternoon periods, and the more trustworthy it is.
Here, we add the concept of parking service completion rate to effectively improve the time decay of trust values. There are records in the cloud server that the number of services each vehicle node has provided and the number of services that are finally completed. For example, for the most recent \( n \) days, the vehicle node \( i \) has provided \( v_{total} \) services during the morning and afternoon parking periods. The service deposited on node \( i \) would be terminated if \( i \) left halfway. Then, those terminated services would be regarded as failing services. Mark the number of failed services as \( v_{false} \), so the number of services completed by node \( i \) in these two time periods is \( v_{total} - v_{false} \). Then, the parking service completion rate of the node \( i \) during these two parking periods can be obtained by formula (3)

\[
\phi(i) = \frac{v_{total} - v_{false}}{v_{total}} \tag{3}
\]

In summary, the parking service completion rate can be expressed as formula (4)

\[
\phi(i, n) = \frac{\sum_{j=1}^{n} v_j}{\sum_{j=1}^{n} V_j} \tag{4}
\]

where \( \phi(i,n) \) represents the parking service completion rate of the vehicle node \( i \) in the last \( n \) days, and \( v_j \) represents the number of services that the vehicle node \( i \) finally completed in the morning time period and the afternoon time period in the \( j \)-th day. \( \sum_{j=1}^{n} v_j \) represents the total number of services that node \( i \) finally completed in those two time periods in the recent \( n \) days. \( V_j \) denotes the number of services provided by node \( i \) during the morning and afternoon time periods in day \( j \). \( \sum_{j=1}^{n} V_j \) denotes the total number of service providing by node \( i \) in those two time periods in the recent \( n \) days. The higher \( \phi(i,n) \) is, the higher reliability of the service providing by node \( i \) during the morning and afternoon time periods, and node \( i \) is more trustworthy in these two time periods. Obviously, the value of \( \phi(i,n) \) is among \((0,1)\).

By combining the above-mentioned parking time coverage and parking service completion rate, the characteristic property value \( Utr(i,n) \) of node \( i \) in the recent \( n \) days is expressed as formula (5)

\[
Utr(i, n) = \alpha_2 \eta(i, n) + \beta_2 \phi(i, n), \quad \text{where } \alpha_2 + \beta_2 = 1, \quad \text{and } 0.4 < \alpha_2 < 0.6 \tag{5}
\]

where \( \alpha_2 \) and \( \beta_2 \) represent the proportion of \( \eta(i,n) \) and \( \phi(i,n) \). As both parking coverage time and the parking service completion rate have obvious correlation with the characteristic value, the value difference of \( \alpha_2 \) and \( \beta_2 \) should not be too large. The recommended values are \( 0.4 < \alpha_2 < 0.6 \) and \( \beta_2 = 1 - \alpha_2 \).

### Calculation of interaction scale value

When the service providing by the vehicle node \( i \) ends, node \( i \) will obtain the interaction scale value \( Ttr(i) \) based on the size of the storage resource it provides and the time length of services. The storage resource is divided into virtual machines in blocks, which can make each piece of storage independent of each other and reduce the damage caused by malicious attacks. Combined with previous history, the size of storage resources provided by a node is expressed as formula (6)

\[
\mu(i, f_n) = \frac{\sum_{p=1}^{f_n} s_p}{f_n} \tag{6}
\]

Here, \( f_n \) represents the frequency that vehicle node \( i \) provides services during those two time periods. The frequency refers to the number of times node \( i \) provides the storage resource service. \( s_p \) represents the number of virtual machines provided by node \( i \) in the \( p \)-th time. \( \sum_{p=1}^{f_n} s_p \) represents the total number of virtual machines provided by node \( i \) after \( f_n \) times services it provides in the recent \( n \) days. \( \mu(i, f_n) \) indicates the average number of virtual machines provided by node \( i \) in each time among those two time periods among \( n \) days.

The time length of services vehicle node \( i \) provides will also be synthesized according to its historical record, which is expressed as formula (7)

\[
\rho(i, f_n) = \frac{\sum_{p=1}^{f_n} t_p}{f_n} \tag{7}
\]

The symbolic description of formula (7) is similar to formula (6), replacing the storage size with the length of time. According to the actual situation, when a vehicle node provides more services in those two periods in the recent \( n \) days, the higher its reliability. The maximum threshold of the interaction frequency is set here to limit the value, which is denoted as \( F_{max} \).

In summary, the interaction scale value \( Ttr(i) \) of node \( i \) can be expressed as formula (8)

\[
Ttr(i, f_n) = \begin{cases} 
\alpha_3 \mu(i, f_n) + \beta_3 \rho(i, f_n) \bigg/ F_{max}, & \text{where } f_n < F_{max} \\
\alpha_3 \mu(i, f_n) + \beta_3 \rho(i, f_n), & \text{where } f_n \geq F_{max}
\end{cases} \tag{8}
\]

Here \( \alpha_3 \) and \( \beta_3 \) represent the proportion of \( \mu(i, f_n) \) and \( \rho(i, f_n) \) in the interaction scale value \( Ttr(i) \), \( \alpha_3 + \beta_3 = 1 \) and \( 0.4 < \alpha_3 < 0.6 \). When \( f_n \geq F_{max} \), \( f_n / F_{max} \) is set to 1. As storage space service and the time duration provided by a node are equally important for the
interaction scale value, value difference between $\alpha_3$ and $\beta_3$ should not be too large.

Considering that $\mu(i, f_n)$ and $\rho(i, f_n)$ represent different physical meanings with different units, simply synthesizing from numerical will have a negative effect on the final result $Tr(i, f_n)$. Here, the minimum–maximum normalization method is adopted to improve the synthesis method.

After normalization, $\mu(i, f_n)$ can be calculated as formula (9)

$$\mu'(i, f_n) = \frac{\sum_{p=1}^{f_n} t_p - s_{\min}}{s_{\max} - s_{\min}}$$

(9)

In the above formula, $s_{\min}$ and $s_{\max}$ represent the minimum and the maximum number of virtual machines provided by node $i$ once a service in the recent $n$ days, respectively. The value of $\mu'(i, f_n)$ is among (0,1) after normalization procession.

After normalization, $\rho(i, f_n)$ can be calculated as formula (10)

$$\rho'(i, f_n) = \frac{\sum_{p=1}^{f_n} t_p - s_{\min}}{s_{\max} - s_{\min}}$$

(10)

In the above formula, $t_{\min}$ and $t_{\max}$ represent the minimum and the maximum service time length provided by node $i$ once a service in the recent $n$ days, respectively. The value of $\rho'(i, f_n)$ is also among (0,1).

In summary, the interaction scale value $Tr(i)$ of the service vehicle node $i$ providing can be finally expressed as formula (11)

$$Tr(i, f_n) = \begin{cases} \alpha_3 \mu'(i, f_n) + \beta_3 \rho'(i, f_n) & \text{if } f_n < F_{\max} \\ \alpha_3 \mu'(i, f_n) + \beta_3 \rho'(i, f_n) & \text{if } f_n \geq F_{\max} \end{cases}$$

(11)

The trust value calculation method described above provides assistance for the screening of vehicle nodes and the evaluation of services in a certain application.

Obviously, the time complexity of this method depends on the number of days $n$ under statistics and the average number of vehicle nodes in the parking lot among $n$ days. And the time complexity of the method is $O(n^2m)$.

Table 1 gives the comparison of trust model based on dynamic incentive mechanism (DI-Trust) and other models such as Eigen-Trust/Power-Trust and Peer-Trust. The Eigen-Trust reputation management system can isolate malicious peers in P2P sharing networks. It uses global trust values and a peer node $i$ will compute and store other peer nodes’ trust values according to its own computation. Peer-Trust model is a kind of coherent adaptive model. It introduces three basic trust parameters and two adaptive factors to compute trust values. The main feature of this trust model is that the feedback obtained from other nodes is used while calculating a node’s trust value.$^{15}$

### Storage resource management and scheduling mechanism

The storage resource management and scheduling mechanism proposed in this article synthesize four factors of a candidate service providing vehicle node. They are the trust value, the parking time coverage, the virtual machine utilization rate, and the number of tasks carried by the node. This article divides storage requirements into three levels: high, medium, and low. According to three different level requests, the mechanism calculates to get a reasonable scheduling scheme. It should be noted that when the value of “last $n$ days” is determined, all calculations adopt the same $n$ value.

For the sake of simplification, the following calculation formula does not include the parameter $n$.

#### High-level event scheduling method

After comprehensively calculating the above four factors, the composite value $W(i)$ of a candidate service providing vehicle node $i$ is calculated as formula (12)

$$W(i) = \frac{(-Dtr(i)^2 + 2Dtr(i) \eta(i))}{\sqrt{Vu(i) + 1} \sqrt{\frac{T_{\max}(i)}{F_{\max}}} + 1}$$

(12)

Here, meanings of symbol $Dtr(i)$ and $\eta(i)$ are the same as in formulas (1) and (2) except that parameter $n$ is ignored. $Vu(i)$ represents the proportion of virtual machines being occupied in the candidate service providing node $i$. $T_{\max}$ represents the number of storage tasks currently carried by node $i$. $T_{\max}$ represents the maximum number of tasks that a node is allowed to undertake. Therefore, $P(i)$ can be calculated as formula
The time complexity of the scheduling method is only related to the vehicle nodes number $m$, and it is $O(m)$.

### Simulation experiments and result analysis

This part introduces some simulation experiments. These experiments focus on aspects of sub-task interruption rate, the load balance of the whole environment, and the change of a vehicle node trust value.

### Simulation environment

Simulation experiments in this article are carried out by MATLAB R2014a. The application scenario of storage resource scheduling on the parking lot is simulated. Suppose there are up to 500 vehicle nodes in the parking lot. The characteristic attributes of a vehicle node include the ID of the vehicle node, the parking position, time of entering the parking lot every day, parking time coverage of four time periods in the recent $n$ days, the total number of virtual machines that the node can afford, the number of virtual machines that have already been occupied of the node, and the value of trust of the node. One simulation experiment is run 10 times and takes the average value as the final result. In the above introduction, the relationship and range of $\alpha$ and $\beta$ are given. On this basis, we determine the value of this parameter according to the best results in the experimental process. The global parameters in simulation experiments are shown as Table 2.

The value range of node trust value is among $[0, 1]$, with 0.4 as the initial trust value of a newly added vehicle node. Simulation experiments set up 300 storage tasks belong to the three level events, and the start time of each task is set. The end time of a task is set randomly. It corresponds to the leaving time of the vehicle node that publishes this task. A service providing node

### Medium-level and low-level event scheduling methods

Medium-level and low-level event scheduling methods are almost the same as the high-level event except the calculation method of $W(j)$

$$W(i) = \frac{Dtr(i) \eta(i)}{\sqrt{Vu(i)} + \sqrt{\frac{T_d(i)}{T_{max}} + 1}}$$ (15)

As medium-level event, the value of $W(j)$ can be calculated according to formula (15). Then, $P(i)$ and $N_{\text{total}}(i)$ can be got according to formulas (13) and (14). If the number of idle virtual machines of a vehicle node is less than the number required by the task, the excess virtual machines will be placed on other nodes one by one from high parking time coverage nodes to low nodes.
vehicle does not know how long will be the task need to last. The proportion of three level event tasks is shown as Table 3.

**Simulation results and performance analysis**

**Sub-task interruption rate.** The departure of a vehicle node that carrying the storage task will result in the forced interruption of the carrying task. As mentioned earlier, a storage task is split into multiple sub-tasks in virtual machines for scheduling. This article studies the variation of sub-task interruption rate under storage resource scheduling method based on DI-Trust model. The parameter settings for the simulation experiment are shown in Table 4.

Here, this article uses a performance indicator: sub-task interruption rate, that is, the ratio of the number of sub-tasks interrupted to the number of total sub-tasks. The lower the sub-task interruption rate is, the less the service providing vehicle nodes leave before the task is finished, and the better service experiences a user gets. The experiment simulates the DI-Trust model–based storage resource scheduling method proposed in this article and compares with the sub-task interrupt rate based on the Peer-Trust model, Eigen-Trust model, and None model. The results are shown in Figures 3–6.

The None model in Figures 3–5 refers to the random designation of vehicle nodes to provide service when there is a task request without using any trust model. It can be seen from figures that the sub-task interruption rate of the None model is much higher than that of other trust models. When the number of tasks is small, the DI-Trust model shows better service performance than other models. As the number of tasks increases, the gap between the three trust models is gradually narrowing in the comparison of sub-task interruption rate. This is because more good status nodes are occupied and choice becomes less as the number of tasks increases. DI-Trust model is always a trust model with good performance of storage resource scheduling in a parking lot.

Figure 6 gives the comparison of sub-task interruption rate of high-, medium-, and low-level event under DI-Trust model. The abscissa of the figure is the number of tasks and the ordinate is the sub-task interruption rate. Each level event is experimented with 150 random tasks. It can be seen from the figure, for high-level event, the sub-task interruption rate keeps low, the change range is smooth, and after 150 tasks are finished, the sub-task interruption rate is still less than 10%. This is in line with our goal of minimizing the interruption rate of high-level event to ensure users’ information security and service experience.

**Load balance among vehicle nodes.** In the scheduling of storage resources, a problem that needs to be

| Table 2. General simulation parameters for simulation experiments. |
| Parameter name | Parameter meaning | Value |
| NUM_CARS | Total number of vehicle nodes | 500 |
| NUM_TASKS | Total number of tasks | 300 |
| TOTAL_STORE | The maximum storage space of a vehicle node | 32 GB |
| $f_{\text{max}}$ | Maximum frequency of interaction | 50 |
| $n$ | Recent days | 15 |
| $\alpha_1$ | Feature attribute scale factor | 0.3 |
| $\beta_1$ | Interaction scale factor | 0.7 |
| $\alpha_2$ | Parking time coverage scale factor | 0.5 |
| $\beta_2$ | Service completion rate scale factor | 0.5 |
| $\alpha_3$ | Average VM scale factor | 0.4 |
| $\beta_3$ | Average service time scale factor | 0.6 |

VM: virtual machine.

| Table 3. Proportion of three level event storage tasks. |
| Event level | High-level event | Medium-level event | Low-level event |
| File size | (0 B,10 MB) | (10 MB,100 MB) | (100 MB,1000 MB) |
| The proportion | 20% | 30% | 50% |

| Table 4. Parameter settings for sub-task interruption rate simulation experiments. |
| Parameter | Meaning of the parameter | Value |
| NUM_CARS | Total number of vehicle nodes | 500 |
| NUM_TASKS | Total number of tasks | 300 |
| TOTAL_STORE | Total vehicle node storage | 32 GB |
| SIZE_VM | A virtual machine block’s size | 64 kB |
| ERROR_DEPA_TIME | Error between estimated departure time and real departure time | ± 30 min |
| TASK_MAX | The maximum number of tasks allowed to be carried by a single vehicle node | 30 |
| POINT_MAX | The maximum number of individual sub-tasks allowed to be divided | 20 |
considered is to make the vehicle node undertake the task equilibrium as far as possible. The standard deviation of space usage is used to evaluate vehicle nodes’ load status. This value can reflect the degree of dispersion of all nodes’ space usage in the parking lot. The lower the standard deviation is, the more balanced the load of nodes is. Parameter settings for this simulation experiment are shown in Table 4. Experiments’ results are shown in Figure 7.

In Figure 7, as the number of tasks increases, the standard deviation of space usage shows a downward trend under the DI-Trust model and the Peer-Trust model. However, when the number of tasks reaches 300, there is a rebound of the standard deviation obtained by the Peer-Trust model. The standard deviation under the DI-Trust model has been always in a decreasing trend, although the trend decreased with the number of tasks increases. In contrast to those two models, the standard deviation keeps increasing as the number of tasks increases under the Eigen-Trust model. Therefore, the DI-Trust model has a more prominent contribution to the improvement of load balance of vehicle nodes in the parking lot.

Figure 8 presents a comparison of vehicle nodes’ space usage before and after task assignment using the DI-Trust model to do scheduling. Tasks in the experiment randomly selects 50 nodes for task carrying. The horizontal coordinates in the figure represent the node sequence number, and the vertical coordinates are the node storage space usage. The spatial usage of 50 nodes varied and fluctuated greatly at the beginning. After 150 storage tasks scheduling finished, the space usage...
among nodes tends to about 70%. It can be seen in the figure that the fluctuation of the space usage rate of each node after carrying the task is greatly reduced (marked with red color) compared with that before the task scheduling (marked with blue color), and the purpose of load balancing is achieved.

**Observation of nodes’ trust value.** In this experiment, three vehicle nodes are randomly selected from the parking lot to observe the variation of their trust values under the DI-Trust model. The characteristic properties of three nodes are shown in Table 5.

In the experiment, three nodes completed a high-level storage service for 10 days. Changes of their trust values over 10 days are shown in Figure 9. The horizontal coordinates in the figure represent days, and the vertical coordinates represent the node trust value. The three nodes are marked with three colors: blue, green, and red.

The task completion rate (not be interrupted) of each node, the average number of virtual machines, and the average service time provided by each node per day are shown in Table 6.

![Figure 8. Comparison of node space usage under the DI-Trust model.](image)

**Table 5.** Characteristic properties of three nodes under trust value experiment.

| Node name | Initial trust value | Parking time coverage in recent n days |
|-----------|---------------------|----------------------------------------|
| H1        | 0.4                 | 94.3%                                  |
| H2        | 0.4                 | 85.7%                                  |
| H3        | 0.4                 | 78.1%                                  |

![Figure 9. Trust values of three nodes over 10 days.](image)

**Table 6.** Observation of key parameters of three nodes.

| Node name | Parking time coverage | Task service completion rate | Average number of virtual machines provided per day | Average daily service time (min) | Initial trust value | Trust value after 10 days |
|-----------|-----------------------|-----------------------------|-----------------------------------------------------|---------------------------------|---------------------|---------------------------|
| H1        | 94.3%                 | 89.4%                       | 289                                                 | 2660                            | 0.4                 | 0.689                      |
| H2        | 85.7%                 | 77.6%                       | 301                                                 | 3335                            | 0.4                 | 0.619                      |
| H3        | 78.1%                 | 67.6%                       | 212                                                 | 2965                            | 0.4                 | 0.562                      |

In Figure 9, the trust values of the three nodes show an increasing trend with the passage of time and the number of services, but as time goes on, the increase is smaller and smaller. The initial trust values of the three nodes are the same. Values start to change after nodes begin to provide service, and the trust value gap among nodes becomes obvious with the increase in providing service.

Combined with the data in Table 6, we can find that the initial gap of parking time coverage of three nodes is obvious, which leads to the difference of service completion rate. The higher the parking time coverage of the node, the lower the probability of the vehicle leaving, and the lower the possibility of interruption during
the service task, so the higher the service completion rate will be correspondingly. It can be seen from the figure that the average number of virtual machines provided by node H2 and the average length of service per day are higher than that of node H1, but in the final trust value comparison, the trust value of H2 is lower than that of H1. Although H2 offers many virtual machines and long service time, its service completion rate is low. And the service completion rate is the focus of concern. That is the reason why the final trust value of H2 is lower than that of H1.

**Conclusion**

This article mainly discusses the storage resource scheduling of vehicle nodes in the parking lot, and proposes a vehicle cloud storage resource scheduling method based on DI-Trust model. The model considers a variety of dynamic elements that will affect the trust value of a vehicle node. Based on the trust model, a virtual machine scheduling method based on storage resource requirement is proposed for virtual machine scheduling among vehicle nodes. Simulation experiments are carried out on the interruption rate of sub-task, the load balancing, and the change of vehicle node trust value. These experiments verify the function and performance of the model.

Because there are some differences between the simulation platform and the actual vehicle cloud environment, the vehicle cloud storage resource scheduling method based on the DI-Trust model proposed in this article also needs to be tested in the real vehicle cloud. On the basis of scheduling storage resources, we can further study other types of resource scheduling of vehicle nodes.

**Acknowledgements**

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Science Foundation of Nantong of Jiangsu Grants (JC2018131) and the National Natural Science Foundation of China (62002179).

**ORCID iD**

Xiang Gu https://orcid.org/0000-0002-2594-4118

**References**

1. Liu H. Research on vehicle collaboration incentive mechanism for vehicle cloud. Beijing, China: Beijing Jiaotong University, 2018.
2. Horng SJ, Lu CC and Zhou W. An identity-based and revocable data-sharing scheme in VANETs. *IEEE T Veh Technol* 2020; 69: 15933–15946.
3. Lekha SJI and Kathiroli R. Trust based certificate revocation of malicious nodes in MANET. In: International conference on advanced communication control & computing technologies, Ramanathapuram, India, 8–10 May 2015, pp.1185–1189. New York: IEEE.
4. Li W and Song H. ART: an attack-resistant trust management scheme for securing vehicular ad hoc networks. *IEEE T Intell Transp Syst* 2016; 17(4): 960–969.
5. Zhang J. Trust management for VANETs: challenges, desired properties and future directions. *Int J Distrib Syst Tech* 2012; 3(1): 48–62.
6. Meng H, Zheng K, Chatzimisios P, et al. A utility-based resource allocation scheme in cloud-assisted vehicular network architecture. In: IEEE international conference on communication workshop, London, 8–12 June 2015, pp.1833–1838. New York: IEEE.
7. Seyed AS, Abdul HA, Wan H, et al. Trust management in vehicular ad hoc network: a systematic review. *EURASIP J Wireless Commun Network* 2015(1): 146.
8. Fan N and Wu CQ. On trust models for communication security in vehicular ad-hoc networks. *Ad Hoc Natw* 2018; 26(1): 1–13.
9. Hasrouny H, Samhat AE, Bassil C, et al. Trust model for secure group leader-based communications in VANET. *Wireless Netw* 2018; 25: 4639–4661.
10. Refaat TK, Kantarci B and Mouftah HT. Dynamic virtual machine migration in a vehicular cloud. In: *IEEE symposium on computers and communications (ISCC)*, Funchal, 23–26 June 2014. New York: IEEE.
11. Meneguette RI, Boukerche A and Pimenta AHM. AVARAC: an availability-based resource allocation scheme for vehicular cloud. *IEEE T Intell Transp Syst* 2018; 20: 3688–3699.
12. Yu R, Huang X, Kang J, et al. Cooperative resource management in cloud-enabled vehicular networks. *IEEE T Ind Electron* 2015; 62(12): 7938–7951.
13. Li L, Zhu A and Su T. Research on trust management for cloud computing. *Comput Modern* 2012(9): 109–114.
14. Lu K, Wang J and Li M. An Eigentrust dynamic evolutionary model in P2P file-sharing systems. *Peer Peer Netw Appl* 2015; 9(3): 599–612.
15. Meng X. SureTrust: a super peer-aware trust model for 2-level P2P networks. *Peer Peer Netw Appl* 2018; 11(3): 619–631.