RESEARCH ARTICLE

An energy-efficient failure detector for vehicular cloud computing

Jiaxi Liu, Zhibo Wu, Jian Dong*, Jin Wu, Dongxin Wen

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

* hitljx@gmail.com

Abstract

Failure detectors are one of the fundamental components for maintaining the high availability of vehicular cloud computing. In vehicular cloud computing, lots of RSUs are deployed along the road to improve the connectivity. Many of them are equipped with solar battery due to the unavailability or excess expense of wired electrical power. So it is important to reduce the battery consumption of RSU. However, the existing failure detection algorithms are not designed to save battery consumption RSU. To solve this problem, a new energy-efficient failure detector 2E-FD has been proposed specifically for vehicular cloud computing. 2E-FD does not only provide acceptable failure detection service, but also saves the battery consumption of RSU. Through the comparative experiments, the results show that our failure detector has better performance in terms of speed, accuracy and battery consumption.

OPEN ACCESS

Citation: Liu J, Wu Z, Dong J, Wu J, Wen D (2018) An energy-efficient failure detector for vehicular cloud computing. PLoS ONE 13(1): e0191577. <https://doi.org/10.1371/journal.pone.0191577>

Editor: Lixiang Li, Beijing University of Posts and Telecommunications, CHINA

Received: May 12, 2017

Accepted: January 8, 2018

Published: January 19, 2018

Copyright: © 2018 Liu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: WAN files are available from the <https://figshare.com/s/5297ddc238766def6afc> database. Wireless data are within the paper and its Supporting Information files.

Funding: This work was supported by the National Natural Science Foundation of China (61100029) (<http://www.nsf.gov.cn/>); Jian Dong is responsible for organizing and implementing the project. This work was also supported by the National Natural Science Foundation of China (61370085) (<http://www.nsf.gov.cn/>); Dongxin

Introduction

Vehicular cloud computing is a promising paradigm that aims at merging mobile cloud computing and vehicular networking, in order to give rise to integrated communication-computing platforms [1]. In vehicular cloud computing, vehicles can be either the service providers to enrich existing cloud services by providing various on-road information (e.g., traffic condition like Pics-on-Wheels proposed by [2]) or be the service consumers to enjoy existing centralized Internet cloud services [3–5]. One of the key features of vehicular cloud computing is high mobility [5–9]. The running job may be interrupted by the random arrival and departure of vehicles [3,8]. We must be able to address the mobility and provide an effective control scheme to guide service conditions and cloud resources [10–11]. Thus fault-tolerant schemes are designed to provide reliable and continuous services in vehicular cloud computing despite the unavailability of some vehicles [12–17]. As an essential building block of the fault-tolerant scheme, a failure detector (FD) plays a critical role in the engineering of such dependable systems [18–19]. An optimized FD should find the vehicles' failure in a timely and accurate manner [20].

To improve connectivity of vehicular cloud computing, the Roadside Unit (RSU) is deployed along the road [21–23]. For example, many future Internet applications [24] which are delay and delay-jitter sensitive (such as Netflix and VTube) are benefit from RSU.

Wen is responsible for organizing and implementing the project.

Competing interests: The authors have declared that no competing interests exist.

Normally, they use the solar as power input due to the unavailability or excess expense of wired electrical power [25–28]. The solar power are easily affected by the natural environment, e.g., solar power cannot be acquired at night or cloudy day. The energy supply of RSU is unstable, so the energy capacity of RSU is limited. According to the U.S. Department of Transportation [29], it is estimated that 40% of all initial rural freeway roadside infrastructure would have to be solar powered by 2050. A breakdown of the deployment costs also found that over 63% of these roadside infrastructure costs would be consumed by solar energy provisioning, e.g., solar panels, batteries, and their associated electronics. Thus, it is important to reduce the energy consumption of RSUs [28,30].

Existing main adaptive FDs (such as Chen FD [31], ϕ -FD [32], ED FD [33] etc.) keep a sliding window (WS) that contains information about received messages to make an estimate of the state (trusted or suspected of having failed) of a monitored node. These FDs need a certain memory space to save a large history message window. At each detection cycle, a large amount of calculation is needed to compute the probability distribution parameters and detector parameters [34]. For most RSUs with battery, these overhead of FDs can exacerbate the battery consumption.

In this paper, aiming at the RSUs with battery, we have presented the Energy-Efficient Failure Detector (2E-FD). It does not rely on the probability distribution of message transmission delay, or on the maintenance of history message windows. We use the arrival time of last message to estimate the arrival time of next message. In addition, the dynamic safety margin, which is computed by the single exponential smoothing method, is used to improve the accuracy of failure detection. To evaluate the performance of 2E-FD, some best-known existing FDs are selected in terms of detection time, mistake rate and query accuracy probability. Besides, we also measure the battery consumption of RSU with different FDs. The experimental results show that 2E-FD is able to reduce the battery consumption of RSU and provide an adaptive failure detection service with high accuracy.

The rest of this paper is organized as follows. In section 2, the related work of vehicular cloud computing and failure detectors is introduced. Section 3 introduces the system model and presents the implementation of 2E-FD. Section 4 carries out the experiments on real traces and tests the battery consumption of RSU with different FDs. Finally, the work is concluded in section 5.

Related work

In this section, vehicular cloud computing is firstly introduced. Second, the quality of service (QoS) metrics of FD is introduced. Finally, several existing main adaptive FDs are presented.

Vehicular cloud computing

Olariu et al. [35–37] advocate the concept of vehicular cloud, which coordinates the computing, sensing, communication and storage resources to provide services to authorized users. Different from conventional Internet cloud with dedicatedly installed hardware, vehicular cloud leverages the already available resources on vehicles. In vehicular cloud computing, vehicles communicate with the data centers where the wanted cloud services locate using vehicle-to-infrastructure (V2I) communications, e.g., LTE, WiMax. In addition, vehicles can also work in an autonomous way solely relying on the vehicle-to-vehicle (V2V) communication capabilities. Accordingly, we think that vehicular cloud can be described by a loose two-tier architecture. The first-tier is the Internet cloud computing platform (e.g., data centers) while the second-tier consists of many vehicular cloudlet [3]. The vehicular cloudlet is made up of

Fig 1. Architecture of vehicular cloud computing.

<https://doi.org/10.1371/journal.pone.0191577.g001>

RSU and vehicles. A user can acquire cloud services from either the first-tier data center or the second-tier vehicular cloudlet. An architecture example is illustrated in Fig 1.

QoS metrics of failure detector

Many distributed applications have some timing constraint on the behaviors of FDs [19,38]. It is not acceptable that a node is suspected hours after than it has crashed or the FD outputs several false positives. To solve this problem, Chen [31] proposed a series of metrics to specify the QoS of FD: how fast it detects actual failures and how well it avoids false detections. These metrics can quantitatively represent the detection speed and accuracy. We use T or S to represent whether a node is trusted or suspected. T -transition means that the output of the detector changes from S to T , while S -transition means that the output of the detector changes from T to S . The following three primary metrics are used to describe the QoS of a FD.

Detection time (T_D) is the time that elapses from the moment when a node crashes to the time when it starts being suspected, i.e., when the final S -transition occurs.

Fig 2. Query accuracy probability and mistake rate.

<https://doi.org/10.1371/journal.pone.0191577.g002>

Mistake rate (λ_M) is the number of mistakes that a FD makes per unit time, i.e., it represents how frequently a FD makes mistakes.

Query accuracy probability (P_A) is the probability that the output of a FD is correct at a random time.

The first metric is related to a failure detector's speed, while the remaining relate to its accuracy. In many cases, the mistake rate is not sufficient to describe the accuracy of a FD; simultaneously, the query accuracy probability is also needed. For example, Fig 2 shows that both FD_1 and FD_2 are detecting the node p . The two FDs have the same mistake rate (0.125) but different query accuracy probabilities (0.75 and 0.5).

Adaptive failure detector

The adaptive FDs are designed to adapt to changing network conditions and application requirements [39]. In most adaptive FDs, their implementations are based on a heartbeat strategy. Existing main adaptive FDs (Chen FD, φ FD and ED FD) work as follows:

Chen et al. [31] proposed the QoS-based adaptive failure detection algorithm based on a probability network model. This algorithm assumes that node p sends heartbeat message m to node q periodically. The recent n heartbeat messages $m_1, m_2, \dots, m_n$ stored in a sliding window at q . $A_1, A_2, \dots, A_n$ are their receipt times according to q 's local clock. Then, the expected arrival time of next heartbeat message is estimated by:

$$EA_{(k+1)} = \frac{1}{n} \sum_{i=k-n+1}^k (A_i - \eta * i) + (k+1)\eta \quad (1)$$

where η is the sending interval, decided by the QoS requirement of user. The freshpoint τ_{k+1} of next heartbeat message is consist of $EA_{(k+1)}$ and constant safety margin α . One has

$$\tau_{k+1} = EA_{(k+1)} + \alpha \quad (2)$$

This FD estimates the arrival time of next heartbeat message based on a constant safety margin.

In φ -FD, the output is a continuous value φ to represent the suspicion level of the monitored node, rather than a binary value simply representing true or suspect. It keeps a sliding window to save the most recent arrival time of heartbeat messages, and assumes that the heartbeat inter-arrival time follows a normal distribution. Then the value of φ can be calculated as follows:

$$\varphi(T_{now}) = -\log_{10}(P_{later}(T_{now} - T_{last})) \quad (3)$$

where the T_{last} is the time when the most recent heartbeat message is received, T_{now} is the

current time, and $P_{later}(t)$ is the probability of a heartbeat message will arrive more than t time units later than the previous one. According to the assumption of heartbeat inter-arrival time, $P_{later}(t)$ is given by the following equation:

$$P_{later}(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_t^{\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 - F(t) \quad (4)$$

where $F(t)$ is the cumulative distribution function of a normal distribution with mean μ and variance σ^2 . When the applications query the φ -FD at time T_{now} , φ FD will return a value of φ to them. Then each application compares the value φ with its threshold Φ , which is given by different QoS requirements. If $\varphi > \Phi$, a certain action is triggered. Thus, φ -FD can meet different QoS requirements of multiple applications simultaneously.

ED FD is similar to the φ -FD. The difference is in the assumption of distribution of heartbeat inter-arrival time. In ED FD, it assumes that the heartbeat inter-arrival time follows an exponential distribution. Then, the suspicion level is given by a value, called e_d , which is calculated as follows:

$$e_d = F(T_{now} - T_{last}) \quad (5)$$

$$F(t) = 1 - e^{-\frac{t}{\mu}} \quad (6)$$

where T_{now} , T_{last} and μ have the same meaning as for the φ FD. For ED FD, the threshold is called E_d .

For a FD, it will belong to the class $\diamond P$ if it satisfies the following properties [40]:

Strong completeness: eventually every node that crashes is permanently suspected by every correct node.

Eventually strong accuracy: there is a time after which a correct node is no longer suspected by any correct node.

Implementation

System model

We consider a partially synchronous system consisting of a finite set of nodes $\Pi = \{p_1, p_2, \dots, p_n\}$. Each node behaves correctly until it crashes and is unable to recover. Any two nodes can be connected by an unreliable communication channel. Because most FDs are implemented using the UDP protocol, we assume that the communication channel between nodes is a fair-lossy channel [41], i.e., no message can be copied or modified and no new message can be created, and if a node p continues sending a message m to q , q will eventually receive m .

We assume the existence of some global time (unknown to nodes), denoted as global stabilized time (GST), and that nodes always make progress; furthermore, at least $\delta > 0$ time units elapse between consecutive steps (the purpose of the latter is to exclude the case where nodes require an infinite number of steps in finite time).

To simplify the description, consider a system that consists of only two nodes p and q , where q is monitoring p . Node p sends a message to q every η time (sending interval) or is subject to crashing. Node q suspects node p if it does not receive any heartbeat message from p for a period of time determined by the freshpoint.

The implementation of 2E-FD

To improve the battery consumption of RSU with failure detector, we propose an energy-efficient failure detector 2E-FD. When a heartbeat message is sent to the receiver, the message

delay d_i can be calculated by

$$d_i = T_{now} - (i - 1) * \eta \quad (7)$$

where η is the sending interval, T_{now} is the arrival time of new heartbeat message. We assume that the message delay of next heartbeat message d_{i+1} is equal to d_i , so the expected arrival time of next heartbeat message is

$$EA_{i+1} = j * \eta + d_i \quad (8)$$

According to single exponential smoothing method [42], for the each d_i , the new predictive delay $\hat{d}_{i+1}$ is computed from the formula:

$$\hat{d}_{i+1} = \alpha * \hat{d}_i + (1 - \alpha) * d_i \quad (9)$$

where α ($0 \leq \alpha \leq 1$) is a constant between 0 and 1, which controls how rapidly the $\hat{d}_{i+1}$ adapts to the delay change. So the safety margin (SM) can be estimated by:

$$SM_{i+1} = \beta |\hat{d}_{i+1} - d_i| \quad (10)$$

where β is a variable, chose such that there is an acceptably small probability that the delay for the heartbeat message will exceed timeout. At last, the freshpoint of heartbeat message ($i + 1$) can be computed by

$$\tau_{i+1} = EA_{i+1} + SM_{i+1} \quad (11)$$

2E-FD is unable to get the communication delay from the sender to the receiver when it is lost. To ensure the effectiveness of the proposed approach, we fill the gaps with a value computed by $d_k = (\frac{k-1}{2}) * \eta + d_l$, where d_k is the estimation of message delay of unreceived heartbeat message m_k from p , d_l is the message delay of receiving last heartbeat message m_l from p .

2E-FD employs the heartbeat approach as the basic failure detection strategy. To simply the description, suppose the system consists of only two nodes p and q , where q is monitoring p . The detection algorithm is shown in Fig 3.

For the node p , it sends heartbeat message to node q at interval $i * \eta$ ($i \geq 0$). For the node q , two tasks will be executed. One task is to suspect the node p when q didn't receive any heartbeat message during the last freshpoint τ of q 's clock. Another task is to compute the freshpoint τ according to the arrival time of new heartbeat message. Every time after receiving the new arrival heartbeat message, node q needs to compute the delay from sending to receiving and safety margin of next heartbeat message.

Correctness proof

From the theory point-of-view, the 2E-FD can satisfy the strong completeness and eventually strong accuracy. Therefore, our FD belongs to class $\diamond P$, which is sufficient to solve the consensus problem. The 2E-FD implements a FD of class $\diamond P$ under the condition of the system model defined in section 3. The evidence is as follows.

According to the system model, there is an upper bound on node speeds and on message transmission times after GST. We assume that Δ_{msg} is the upper bound and node p will not send any heartbeat message after it crashes. The heartbeat message m_{k-1} is the last heartbeat message from p before p crashes. So node q can calculate the freshpoint τ_k of next heartbeat message based on Eq (11).

$$\tau_k = EA_k + SM_k \quad (12)$$

Initialization:

η : sending interval;
 T_{now} : the current time;
 sn_i : the sequence number of last heartbeat message;
 $suspectlist[] \leftarrow \perp$; /*empty suspect list*/

Node p (Sender):

For all $i \geq 1$, at time $(i \cdot \eta)$
 Send heartbeat message m_i to q ;

Node q (Receiver):

Task 1:

If $p \notin suspectlist$, and q didn't receive message during the
 last *freshpoint* τ of q 's clock;
 Add p to *suspectlist*; /*Suspect p */

Task 2:

Upon receiving heartbeat message m_j from p
 If $j > sn_i$
 { If $p \in suspectlist$
 Remove p from *suspectlist*;
 $\beta = \beta + 1$;
 else
 $\beta = 1$;
 /*compute the delay from sending to receiving*/
 $d_j = T_{now} - (j - 1)\eta$;
 $EA_{j+1} = j * \eta + d_j$;
 $\hat{d}_{j+1} = \alpha * \hat{d}_j + (1 - \alpha) * d_j$;
 $SM_{j+1} = \beta |\hat{d}_{j+1} - d_j|$;
 /*compute timeout based on network variation*/
 $\tau_{j+1} = EA_{j+1} + SM_{j+1}$;
 $sn_i \leftarrow j$;
 }

Fig 3. 2E-FD algorithm.

<https://doi.org/10.1371/journal.pone.0191577.g003>

Because $EA_k = (k - 1) \cdot \eta + d_{k-1}$ and $d_{k-1} < \Delta_{msg}$, one gets

$$EA_k < (k - 1) \cdot \eta + \Delta_{msg} \quad (13)$$

For the safety margin $SM_k = \beta |\hat{d}_k - d_{k-1}|$, we have

$$\hat{d}_{k+1} = \alpha \cdot \hat{d}_k + (1 - \alpha) \cdot d_k \quad (14)$$

By recursion property of equation, one gets

$$\begin{aligned} \hat{d}_{k+1} &= \alpha(\alpha \hat{d}_{k-1} + (1 - \alpha)d_{k-1}) + (1 - \alpha)d_k \\ &= \alpha^2 \hat{d}_{k-1} + \alpha(1 - \alpha)d_{k-1} + (1 - \alpha)d_k \\ &\dots \\ &= \alpha^{k+1} \hat{d}_0 + \alpha^k(1 - \alpha)d_0 + \alpha^{k-1}(1 - \alpha)d_1 + \dots + (1 - \alpha)d_k \end{aligned} \quad (15)$$

Because $d_0 = \hat{d}_0 = 0$,

$$\begin{aligned} \hat{d}_{k+1} &= \alpha^{k-1}(1 - \alpha)d_1 + \alpha^{k-2}(1 - \alpha)d_2 + \dots + (1 - \alpha)d_k \\ &= (1 - \alpha) \sum_{i=1}^k \alpha^{k-i} d_i \end{aligned} \quad (16)$$

Therefore, using the same methods, we get

$$\hat{d}_k = (1 - \alpha) \sum_{i=1}^{k-1} \alpha^{k-i} d_i \quad (17)$$

Because $0 \leq d_i < \Delta_{msg}$, therefore

$$0 \leq \hat{d}_k < \alpha(1 - \alpha^{k-1})\Delta_{msg} \quad (18)$$

From inequalities (18), we get

$$SM_k < \beta |\alpha(1 - \alpha^{k-1})\Delta_{msg}| = \beta \cdot \alpha(1 - \alpha^{k-1})\Delta_{msg} \quad (19)$$

Thus, the freshpoint τ_k of next heartbeat message is

$$\tau_k = EA_k + SM_k < (k - 1) \cdot \eta + \Delta_{msg} + \beta \cdot \alpha(1 - \alpha)\Delta_{msg} \quad (20)$$

From inequation (20), all components of τ_k are bounded, so we can deduce τ_k is bounded. If for each message m_{k-1} received from node p , node q activates a bounded timeout, then there is a time after which q suspects p , if it receives no new message from p . Thus the 2E-FD satisfies the strong completeness property.

Every time q time out and p is correct, then β is increased. There is a time t_{bound} where safety margin SM is larger than Δ_{msg} . The heartbeat message from node p must be received by node q . As a result, node q can avoid false detection, and SM stops increasing. Thus, when SM becomes higher than Δ_{msg} , a correct node is no longer suspected by any correct node. This indicates the 2E-FD satisfies the eventually strong accuracy property.

Evaluation and performance

In this section, we will evaluate and analyze the performance of 2E-FD through comparative experiments. First, the 2E-FD was compared with the failure detection algorithms in terms of

detection time, mistake rate and query accuracy probability. Then, we tested the battery consumption of RSU with different failure detection algorithms in a simulation environment.

2E-FD performance evaluation and comparison

To enhance the authenticity of the experiment, we referred to the method in paper [43] that used the same trace files to replay different schemes of FDs and calculated the QoS metrics. This method ensures that all schemes of FDs are compared in the same network condition. The trace files are obtained from two network conditions (Wireless and WAN). For the accrual FDs (ϕ -FD and ED FD), it is necessary to compute the parameters of FD based on multiple history messages. In these experiments, ϕ -FD and ED FD shared the same fixed window size (1000), while the Chen FD used two different sliding windows ($WS = 1$ and $WS = 1000$). Furthermore, we did not analyze the sampled data until the sliding window was full, since the behavior of the FDs is stable only after that moment. The parameters of FDs are configured as follows: the

Fig 4. Mistake rate vs. detection time in Wireless.

<https://doi.org/10.1371/journal.pone.0191577.g004>

basic parameters of 2E-FD are set as $\alpha = 0.85$, and in order to find the best QoS and compare with the others, here $\beta \in [10^{-6}, 10^6]$. For ϕ -FD, the parameters are set the same as in [32]: $\Phi \in [0.5, 16]$. For Chen FD, the parameters are set the same as in [31]: $\alpha \in [0, 10000]$. For ED FD, the parameters are set the same as in [33]: $E_d \in [10^{-4}, 10]$.

For the Wireless case, two nodes are selected to represent the detecting node and monitored node respectively. The two nodes communicated through a WiFi (802.11g) network. The detecting node was equipped with a 900MHz ARM Cortex A7 processor, 1G RAM and Cent OS 6.5 operating system. While the monitored node was equipped with a 2GHz Intel Xeon processor, 1G RAM and Cent OS 6.5 operating system. The heartbeats were sent with a target of one heartbeat every 100.5ms (standard deviation: 7.87ms; min.: 0.002ms; max.: 948.96ms). During the experiment, the round-trip time (RTT) was measured to be at a low rate. The average RTT was 1.83ms, with a minimum of 1.175ms, and a maximum of 21.953ms. More than 1 million heartbeats were sent.

Fig 5. Query accuracy probability vs. detection time in Wireless.

<https://doi.org/10.1371/journal.pone.0191577.g005>

Fig 4 shows the results of the mistake λ_m vs. detection time T_D in the Wireless scenario. The x-coordinate represents the detection time, and y-coordinate represents the mistake rate. Fig 5 shows the results of query accuracy probability P_A vs. detection time T_D in the same scenario.

From the Figures, we found that all the FDs follow the same general tendency. However, our FD outperforms the others in the Wireless scenario. This improvement is because most late heartbeats were caught by the freshpoint under the same network conditions. In Fig 4, when $T_D < 0.2s$, the mistake rate of 2E-FD has obvious decrease than other FDs. This is because that our FD can quickly adapt to the network conditions than other FDs. From Fig 5, our FD has higher query accuracy probability than others. When $0.15s < T_D < 0.4s$, our FD and ED-FD seem to have similar query accuracy probability. At the same time, it is clear that the performance of Chen FD with $WS = 1$ is better than Chen FD with $WS = 1000$ in terms of mistake rate and query accuracy probability. This may be because that many burst data and too old data affect the calculation of freshpoint when sliding window size increases.

Fig 6. Mistake rate vs. detection time in WAN.

<https://doi.org/10.1371/journal.pone.0191577.g006>

For the WAN case, the experiment was carried out on two nodes respectively located in Japan and Switzerland [32]. In the experiment, all heartbeat messages were transmitted using UDP protocol, and the monitored node p sent heartbeat messages to node q at the interval of 100ms, and q recorded all the arriving time of the received messages into the trace file according to its local clock. The experiment lasted a week, and more than 5 million heartbeat messages has been collected, where the average RTT is 283.3ms and message loss rate is approximately 0.4%.

Figs 6 and 7 show the results of the mistake rate λ_m and query accuracy probability P_A vs. detection time T_D in the WAN scenario. Similar to the result in the Wireless scenario, the mistake rate and query accuracy probability of all of the FDs have an identical tendency with increasing detection time. In Fig 6, the 2E-FD presents the lowest mistake rate (an improvement of up to 50%). It shows that our predictive method can improve the detection accuracy effectively. In Fig 7, our FD has an obvious improvement compared with ED FD when $T_D <$

Fig 7. Query accuracy probability vs. detection time in WAN.

<https://doi.org/10.1371/journal.pone.0191577.g007>

0.35s. Furthermore, when the mistake rate or query accuracy probability is the same, our FD performs shorter detection time. In the WAN, 2E-FD behaves better than the others FDs in terms of quick speed and high accuracy.

Comparison of battery consumption

RSUs are static devices that disseminate the data stored in the infrastructure to the passing-by vehicles periodically [28]. In addition, RSU may exploit Infrastructure-to-Vehicular single-hop IEEE 802.11-like wireless links for data dissemination [11]. Based on these, we simulated the working environment of RSU by organizing several nodes. These nodes were connected by wireless link. One node was selected to represent the RSU, and the other nodes represented the vehicles. The RSU, which was equipped with 500mAh battery, was responsible for disseminating data to vehicles [25,28]. The vehicles may be failure via the fault injection method. At each experiment, we deployed different FD to the RSU and measured the running time of RSU.

For the accrual FDs (ϕ -FD and ED FD) and Chen FD, they need to calculate the detector parameters and maintain the history window of detection messages at every detection period. We have selected different sliding window size setting (from $WS = 100$ to $WS = 10000$) and have made a detailed comparison of battery consumption of RSU. To make the experiment general, we re-did the experiments 5 times with the same environment, and the same parameters for each failure detection algorithms. Finally, we recorded the running time of RSU with different failure detection algorithms (as shown in Fig 8).

From the Fig 8, it shows that the running time of RSU without any FD is the longest and improved by 6% compared to the RSU with 2E-FD. For the RSU with FD, the running time of RSU with ϕ -FD is the shortest, and it quickly decreases as the sliding window size changes. This is because that the battery consumption of RSU is exacerbated when the parameters of normal distribution model are calculated. The calculation of parameters needs the statistical data from the entire window. While the running time of RSU with 2E-FD is the longest (an improvement of up to 12% than ϕ -FD when the sliding window size is 10000), and it is not affected by sliding window size. For the RSU with Chen FD ($WS = 1$), its running time is not affect by sliding window size too. However, its running time is shorter than the RSU with

Fig 8. Running time of RSU vs. sliding window size for different FDs.

<https://doi.org/10.1371/journal.pone.0191577.g008>

2E-FD. This might be because that the accuracy of 2E-FD is higher than Chen FD ($WS = 1$) due to the dynamic safety margin. 2E-FD avoids sending more data to the failure nodes so as to save the battery consumption of RSU. The RSU in the experiment shown in Fig 8 at most connects 10 nodes. In vehicular cloud computing, in order to maintain the connectivity of systems, each RSU is required to connect many vehicles. Therefore, the fact that 2E-FD can reduce battery consumption of RSU is more significant in real systems.

Conclusion

Failure detection plays a very important role in vehicular cloud computing. In this paper, we introduced the energy-efficient failure detector (2E-FD). It has been proven that 2E-FD is a failure detector of class $\diamond P$. By using the prediction of last message and the dynamic safety margin, the 2E-FD can quickly adapt to the network conditions and provide acceptable QoS failure detection service. Moreover, 2E-FD is able to save the battery consumption of RSU because it does not need to compute the distribution parameters and maintain the sliding window. Through the comparative experiments, the results showed that 2E-FD demonstrates a better performance in terms of detection speed, accuracy and battery consumption. Therefore, 2E-FD is suitable to be layout in the vehicular cloud computing for providing failure detection service.

Supporting information

S1 File. Wireless.
(ZIP)

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61100029 and 61370085).

Author Contributions

Conceptualization: Jiayi Liu, Jian Dong.

Data curation: Jiayi Liu, Jin Wu.

Formal analysis: Jiayi Liu.

Funding acquisition: Jian Dong, Dongxin Wen.

Investigation: Jiayi Liu.

Methodology: Jiayi Liu.

Project administration: Jiayi Liu.

Resources: Jiayi Liu, Jian Dong, Jin Wu.

Software: Jiayi Liu, Jin Wu.

Supervision: Jiayi Liu, Zhibo Wu, Jian Dong, Dongxin Wen.

Validation: Jiayi Liu, Zhibo Wu, Dongxin Wen.

Writing – original draft: Jiayi Liu.

Writing – review & editing: Jiayi Liu, Zhibo Wu, Jian Dong.

References

1. Whaiduzzaman M, Sookhak M, Gani A, Buyya R. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*. 2014 Apr.; 40(1): 325–344. <https://doi.org/10.1016/j.jnca.2013.08.004>
2. Gerla M, Weng JT, Pau G. Pics-on-wheels: Photo surveillance in the vehicular cloud. *Proceedings of 2013 International Conference on Computing, Networking and Communications (ICNC)*; 2013 Jan. 28–31; San Diego, USA: IEEE, 2013. p. 1123–1127. 10.1109/ICCNC.2013.6504250.
3. Gu L, Zeng D, Guo S. Vehicular cloud computing: A survey. *Proceedings of 2013 Globecom Workshops (GC Wkshps)*; 2013 Dec. 9–13; Atlanta, USA: IEEE, 2013. p. 403–407. 10.1109/GLOCOMW.2013.6825021.
4. Mekki T, Jabri I, Rachedi A, Ben JM. Vehicular cloud networks: Challenges, architectures, and future directions. *Vehicular Communications*. 2016 Dec. 23. <https://doi.org/10.1016/j.vehcom.2016.11.009>
5. Yu R, Huang X, Kang J, Ding J, Maharjan S, Gjessing S, et al Cooperative resource management in cloud-enabled vehicular networks. *IEEE Transactions on Industrial Electronics*, 2016 Sep. 24, 62(12): 7938–7951. <https://doi.org/10.1109/TIE.2015.2481792>
6. Refaat TK, Kantarci B, Mouftah HT. Dynamic virtual machine migration in a vehicular cloud. *Proceedings of 2014 IEEE Symposium on Computers and Communication (ISCC)*; 2014 June 23–26; Funchal, Portugal: IEEE, 2014. p. 1–6. 10.1109/ISCC.2014.6912651.
7. Abhale A, Khandelwal MS, Nagaraj U. Shifting VANET to Cloud—Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2013 Nov. 3, 3(11): 1056–1061.
8. Ghazizadeh P, Mukkamala R, El-Tawab S. Scheduling in vehicular cloud using mixed integer linear programming. *Proceedings of first international workshop on Mobile sensing, computing and communication*; 2014 Aug. 11–11; Philadelphia, USA: ACM, 2014. p. 7–12. 10.1145/2633675.2633681.
9. Refaat TK, Kantarci B, Mouftah HT. Virtual machine migration and management for vehicular clouds. *Vehicular Communications*. 2016 Apr. 5, 4: 47–56. <https://doi.org/10.1016/j.vehcom.2016.05.001>
10. Wei W, Fan X, Song H, Fan X, Yang J. Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud Computing. *IEEE Transactions on Services Computing*. 2016 Feb. 29: 1–1. <https://doi.org/10.1109/TSC.2016.2528246>
11. Shojafar M, Cordeschi N, Baccarelli E. Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Transactions on Services Computing*. 2016 Apr. 7: 1–1. <https://doi.org/10.1109/TSC.2016.2551747>
12. Deng J, Huang SCH, Han YS, Deng JH. Fault-tolerant and reliable computation in cloud computing. *Proceedings of 2010 GLOBECOM Workshops (GC Wkshps)*; 2010 Dec. 6–10; Miami, USA: IEEE, 2010. p. 1601–1605. 10.1109/GLOCOMW.2010.5700210.
13. Jayasinghe D, Pu C, Oliveira F, Rosenberg F, Eilam T. Aeson: A model-driven and fault tolerant composite deployment runtime for iaas clouds. *Proceedings of 2013 International Conference on Services Computing (SCC)*; 2013 June 28– July 3; Santa Clara, USA: IEEE, 2013. p. 575–582. 10.1109/SCC.2013.102.
14. Malik S, Huet F. Adaptive fault tolerance in real time cloud computing. *Proceedings of 2011 IEEE World Congress on Services (SERVICES)*; 2011 July 4–9; Washington DC, USA: IEEE, 2011. p. 280–287. 10.1109/SERVICES.2011.108.
15. Park J, Yu H, Chung K, Lee E. Markov chain based monitoring service for fault tolerance in mobile cloud computing. *Proceedings of 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)*; 2011 Mar. 22–25; Singapore, Singapore: IEEE 2011. p. 520–525. 10.1109/WAINA.2011.10.
16. Ghazizadeh P, Olariu S, Zadeh AG, El-Tawab S. Towards fault-tolerant job assignment in vehicular cloud. *Proceedings of 2015 IEEE International Conference on Services Computing (SCC)*; 2015 June 27– July 2; New York, USA: IEEE, 2015. p. 17–24. 10.1109/SCC.2015.13.
17. Ye T, Su Z, Wu J, Guo L, Li J, Li G. A Safety Resource Allocation Mechanism against Connection Fault for Vehicular Cloud Computing. *Mobile Information Systems*. 2016 Jan. <https://doi.org/10.1155/2016/2747808>
18. Takeuchi K, Tanaka T, Yano T. Asymptotic analysis of general multiuser detectors in MIMO DS-CDMA channels. *IEEE Journal on Selected Areas in Communications*. 2008 Apr. 3, 26(3): 486–496. <https://doi.org/10.1109/JSAC.2008.080407>
19. Tomsic A, Sens P, Garcia J, Arantes L, Sopena J. 2W-FD: A failure detector algorithm with qos. *Proceedings of 30th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2015)*; 2015 May 25–29; Chicago, USA: IEEE; 2015. p. 885–893. 10.1109/IPDPS.2015.74.

20. Lavinia A, Dobre C, Pop F, Cristea V. A failure detection system for large scale distributed systems. Proceedings of 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010); 2010 Apr. 15; Krakow, Poland: IEEE, 2010. p. 482–489. 10.1109/CISIS.2010.29.
21. Atallah R, Assi C, Yu JY. A Reinforcement Learning Technique for Optimizing Downlink Scheduling in an Energy-Limited Vehicular Network. IEEE Transactions on Vehicular Technology. 2016 Oct. 26: 4592–4601. <https://doi.org/10.1109/TVT.2016.2622180>
22. Hammad AA, Todd TD, Karakostas G, Zhao D. Downlink traffic scheduling in green vehicular roadside infrastructure. IEEE Transactions on Vehicular Technology. 2012 Nov. 16, 62(3): 1289–1302. <https://doi.org/10.1109/TVT.2012.2227071>
23. Zou F, Zhong J, Wu W, Du DZ, Lee J. Energy-efficient roadside unit scheduling for maintaining connectivity in vehicle ad-hoc network. Proceedings of 5th International Conference on Ubiquitous Information Management and Communication; 2011 Feb. 21–23; Seoul, Korea: ACM, 2011. p. 64. 10.1145/1968613.1968691.
24. Karagiannis G, Altintas O, Ekici E, Heijenk G, Jarupan B, Lin K, et al. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. IEEE communications surveys and tutorials. 2011 July 12, 13(4): 584–616. <https://doi.org/10.1109/SURV.2011.061411.00019>
25. Azimifar M, Todd TD, Khezrian A, Karakostas G. Vehicle-to-vehicle forwarding in green roadside infrastructure. IEEE Transactions on Vehicular Technology. 2015 Feb. 10, 65(2): 780–795. <https://doi.org/10.1109/TVT.2015.2402177>
26. Khezrian A, Todd TD, Karakostas G, Azimifar M. Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units. IEEE Transactions on Vehicular Technology. 2015 June 27, 64(5): 1942–1957. <https://doi.org/10.1109/TVT.2014.2333665>
27. Atallah R, Khabbaz M, Assi C. Energy harvesting in vehicular networks: a contemporary survey. IEEE Wireless Communications. 2016 May 2, 23(2): 70–77. <https://doi.org/10.1109/MWC.2016.7462487>
28. Qian J, Jing T, Huo Y, Li H, Li Z. Energy-efficient data dissemination strategy for roadside infrastructure in VCPS. EURASIP Journal on Wireless Communications and Networking. 2016 June 16, (1): 1–13. <https://doi.org/10.1186/s13638-016-0650-0>
29. Peirce S, Mauri R. Vehicle-infrastructure integration (VII) initiative benefit-cost analysis: Pre-testing estimates. Intell. Transp. Syst. Joint Program Office, U.S. Dept. Transp., Washington, DC., USA, 2007 Mar.
30. Suryawanshi R, Mandlik G. Focusing on mobile users at the edge of internet of things using fog computing. Int. J. Sci. Eng. Tech. Res. 2015, 4(17): 3225–3231.
31. Chen W, Toueg S, Aguilera MK. On the quality of service of failure detectors. IEEE Transactions on computers. 2002 Aug. 7, 51(5), 561–580. <https://doi.org/10.1109/12.980014>
32. Hayashibara N, Defago X, Yared R, Katayama T. The ϕ accrual failure detector. Proceedings of 23rd IEEE International Symposium on Reliable Distributed Systems; 2004 Oct. 18–20; Florianopolis, Brazil: IEEE, 2004. p. 66–78. 10.1109/RELDIS.2004.1353004.
33. Xiong N, Defago X. ED FD: Improving the Phi accrual failure detector. JAIST, 2007.
34. Ren X, Dong J, Liu H, Li Y, Yang X. Low-Overhead Accrual Failure Detector. Sensors. 2012 May 4, 12(5): 5815–5823. <https://doi.org/10.3390/s120505815> PMID: 22778614
35. Olariu S, Khalil I, Abuelela M. Taking VANET to the clouds. International Journal of Pervasive Computing and Communications. 2011 Apr. 5, 7(1): 7–21. <https://doi.org/10.1108/17427371111123577>
36. Arif S, Olariu S, Wang J, Yan G, Yang W, Khalil I. Datacenter at the airport: Reasoning about time-dependent parking lot occupancy. IEEE Transactions on Parallel and Distributed Systems. 2012 Jan. 31, 23(11): 2067–2080. <https://doi.org/10.1109/TPDS.2012.47>
37. Olariu S, Hristov T, Yan G. The next paradigm shift: from vehicular networks to vehicular clouds. Mobile Ad Hoc Networking: Cutting Edge Directions. 2nd Edition. 2013. <https://doi.org/10.1002/9781118511305.ch19>
38. Xiong N, Vasilakos AV, Wu J, Yang YR, Rindos A, Zhou Y, et al. A self-tuning failure detection scheme for cloud computing service. Proceedings of 26th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2012); 2012 May 21–25; Shanghai, China: IEEE, 2012. p. 668–679. 10.1109/IPDPS.2012.126.
39. Lavinia A, Dobre C, Pop F, Cristea V. A failure detection system for large scale distributed systems. Proceedings of 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2010); 2010 Apr. 15; Krakow, Poland: IEEE, 2010. p. 482–489. 10.1109/CISIS.2010.29.
40. Chandra TD, Toueg S. Unreliable failure detectors for reliable distributed systems. Journal of the ACM (JACM). 1996 Mar. 4, 43(2), 225–267. <https://doi.org/10.1145/226643.226647>

41. Felber P, Défago X, Guerraoui R, Oser P. Failure detectors as first class objects. Proceedings of 1st International Symposium on Distributed Objects and Applications; 1999 Sept. 6–6; Edinburgh, UK: IEEE, 1999. p. 132–141. 10.1109/DOA.1999.794001.
42. Hyndman RJ, Koehler AB, Snyder RD, Grose S. A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting. 2002 Feb., 18(3): 439–454. [https://doi.org/10.1016/S0169-2070\(01\)00110-8](https://doi.org/10.1016/S0169-2070(01)00110-8)
43. Xiong N, Vasilakos AV, Yang LT, Song L, Pan Y, Kannan R, et al. Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems. IEEE Journal on Selected Areas in Communications. 2009 May 5, 27(4). <https://doi.org/10.1109/JSAC.2009.090512>