IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 27, 2019, accepted October 11, 2019, date of publication October 21, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948599

A Hierarchical Failure Detector Based on Architecture in VANETs

JIAXI LIU, FEI DING, AND DENGYIN ZHANG

Jiangsu Key Laboratory of Broadband Wireless Communication and Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

School of Internet of Things, Nanjing University of Posts and Telecommunication, Nanjing 210003, China

Corresponding author: Dengyin Zhang (hitljx@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61571241 and Grant 61872423, in part by the Ministry of Education-China Mobile Research Foundation under Grant MCM20170205, in part by the Scientific Research Foundation of the Higher Education Institutions of Jiangsu Province, China, under Grant 17KJB510043, and in part by the Nanjing University of Posts and Telecommunications Start Foundation (NUPTSF) under Grant XK0160919134.

ABSTRACT The failure detector is one of the fundamental components for maintaining high availability of Vehicular Ad-hoc Networks (VANETs). However, the dynamic nature of VANETs caused by the high mobility of vehicles and communication link failures has a serious impact on the performance of failure detection. Therefore, it is very meaningful to design a suitable failure detector that can deal with the dynamic nature of VANETs well. In this paper, we propose a hierarchical failure detector based on the architecture of VANETs. This failure detector can adapt to the dynamic network conditions and meet the different Quality of Service (QoS) requirements of multiple applications in VANETs. Different from existing failure detectors, we propose a failure detector that employs a detection-result sharing mechanism and groups the nodes according to the architecture of VANETs. We evaluate our proposed failure detector by using NS2 and GT-ITM to simulate the work environment of VANETs. The experimental result shows that our proposed failure detector can improve the detection time by at most 45% and the detection accuracy by at most 25% under similar detection overhead.

INDEX TERMS VANETs, hierarchical failure detection, architecture, QoS.

I. INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) are dynamic, non-structured, self-organizing networks with asynchronous and distributed characteristics, and the nodes (vehicles) move at high speeds compared to other mobile ad-hoc networks (MANETs) [1]–[3]. The main purpose of VANETs is to provide a medium for intervehicular communication allowing for intervehicle (V2V) and vehicle to roadside infrastructure (V2R) data exchange, with multiple applications for Intelligent Transportation Systems (ITS). One of the important features is the high mobility of vehicles in VANETs; vehicles can suddenly quit or enter the network, and communication links among vehicles may suffer from signal degradation due to obstacles, changes in vehicle densities, etc. [4]–[6]. For the ITS applications running on VANETs, VANETs must be fault-tolerant to mitigate communication problems among network nodes so that decisions can be made safely and with confidence [7].

The failure detector plays a key role in a fault-tolerant system [8]. By periodically detecting the status of nodes in systems, it can provide information to achieve routing recovery, application deployment, real-time communication, etc. Thus, the high availability of VANETs can be guaranteed. An effective failure detector can provide information regarding suspected nodes in a timely and accurate manner so that correction actions can be performed as soon as possible. At present, there are numerous studies related to failure detectors in distributed systems [9]–[13]. However, these failure detection algorithms assume that the change of system topology is slow and that the network behavior follows some stable probability distribution in terms of message delay and message loss, and thus they are not adequate for the fast-changing configuration of VANETs.

In this paper, we present a hierarchical failure detector (VC-FD) based on the architecture of VANETs. In VC-FD, the vehicles are divided into different groups according to the Roadside Units (RSUs). Vehicles will share their detection messages with other vehicles and RSU in the same group. Meanwhile, RSUs can exchange messages regarding the status of vehicles in their groups to implement global failure detection. VC-FD can adapt well to the high mobility of vehicles and address the detection accuracy impact of communication link failures in VANETs. Experimental results

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Tedesco.

demonstrate that VC-FD has better performance than the existing failure detector in VANETs. The main contributions of this paper are the following.

1. In the hierarchical failure detection, the function relationships are established between detection parameters and Quality of Service (QoS) metrics so that quantitative output of QoS can be realized.

2. By sharing messages among vehicles, communication link failures can be overcome, and detection accuracy is further improved.

3. Because of the existence of grouping, the detection speed between vehicles is obviously improved.

The rest of this paper is organized as follows. In section 2, the related work regarding VANETs and failure detection is introduced. Section 3 introduces the system model and presents the implementation of VC-FD. The simulation results are reported in section 4. Finally, the work is concluded in section 5.

II. RELATED WORK

In this section, the architecture of VANETs is first introduced. Second, several existing hierarchical failure detectors are presented. Finally, the QoS metrics of the failure detector are introduced.

A. ARCHITECTURE OF VANETs

VANETs are considered to be a subgroup of MANETs in which all nodes are vehicles that move at various speeds [14]. The main objective of VANETs is to enable communication among vehicles on the road and between vehicles and RSUs. For this communication to be possible, devices known as On-Board Units (OBUs) and RSUs must be placed on each vehicle and road, respectively. These devices can send or receive data to or from RSUs. Nevertheless, if a vehicle cannot directly send its data to an RSU, it can relay its data to other vehicles until the data reach an RSU using a multihop transmission strategy.

VANET communication can be categorized into intervehicular communication and vehicle to infrastructure communication. Intervehicular communication refers to the type of communication in which vehicles communicate with each other via wireless technology, also referred to as V2V communication, as shown in Fig. 1. As Fig. 1 illustrates, when a vehicle obtains road condition information, the vehicle immediately begins the information dissemination process using the broadcast communication mode. The vehicles that are near to the vehicle with the information retransmit the message. In this manner, vehicles are notified and can take alternative routes, avoiding a possible traffic congestion problem.

The second mode of communication refers to communication where vehicles and RSUs exchange information. This communication mode is referred to as V2R communication. V2R is the direct wireless exchange of relevant information between vehicles and communication units placed on the sides of roads and avenues. Fig. 1 shows a representation of this type of communication. In this scenario, we observe that,

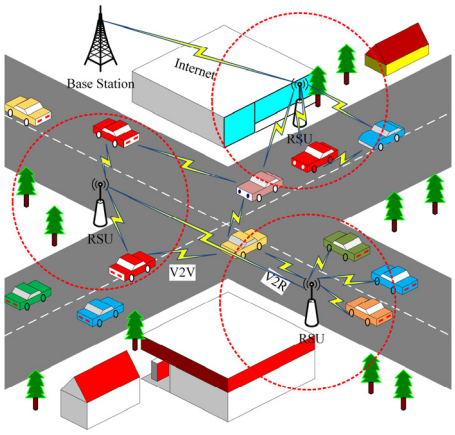

FIGURE 1. Architecture of VANETs.

when a vehicle obtains road condition information, the vehicle begins communication with the RSU to notify it of the problem. The RSU notifies the vehicles that are within its coverage area about the identified problem. At the same time, the RSU can begin the inter-roadside communication process to extend the area of coverage. In this manner, vehicles further away are notified and can take alternative routes, avoiding a potential traffic congestion problem.

B. HIERARCHICAL FAILURE DETECTION ALGORITHM

Hierarchical failure detection algorithms can improve the performance of failure detection by sharing the detection messages among different nodes. Usually, hierarchical failure detection algorithms can be applied to systems that have a special topology or may instantiate such a topology by some strategy. Felber *et al.* [15] proposed a hierarchical failure detection algorithm according to the IP address. In this algorithm, all of the nodes are divided into different LANs. In each LAN, one or several failure detectors can keep track of the states of all local detected nodes and transmit status information to remote monitors in other LANs, thus reducing the number of costly inter-LAN requests. However, this algorithm relies heavily on system topology and therefore does not apply to systems with frequent topology changes. Bertier *et al.* [16] proposed a true two-tier architecture for the failure detection protocol. In this protocol, the system is composed of local groups and a global group. Each group is a detection space, which means that every group member watches all of the other members of its group. Every local group elects exactly one leader that will participate in the global group. Yang *et al.* [17] improved Bertier's failure detection protocol by introducing weights. In the new protocol, the detection frequencies of nodes in the same group are different according to the weights of nodes. The weight of a node is determined by its communication capability and the value of its information assets. However, the two-tier failure detection protocol needs to consume more computing resources to implement grouping. Apolonia [18] proposed a multitier failure detection protocol according to the architecture of cloud computing. This protocol enables monitoring to be performed locally without the need for significant

additional computing resources. Zhuang *et al.* [19] summarized the problem of failure detection between neighboring nodes in overlay networks. Failure detection algorithms are divided into different categories according to different types of detection messages (positive messages and negative messages) and whether neighboring nodes participate in detection. Through theoretical analysis and experimental verification, it is found that a failure detection algorithm that shares two types of detection messages and has neighboring nodes that participate in detection can obviously improve the detection accuracy and speed. The above hierarchical failure detection algorithms assume that the change of system topology is slow and that the network behavior follows some stable probability distribution in terms of message delay and message loss, and thus they are not adequate for the fast-changing configuration of VANETs.

Salvador *et al.* [20] proposed a hierarchical management architecture based on vehicular delay-tolerant networks, which implemented a hierarchical management topology. In this architecture, different groups of nodes are detected by local failure detector with PULL model failure detection strategy. Khatkar *et al.* [21] proposed a fault tolerant scheme, which the RSUs is responsible for monitoring the faulty vehicles in the network. In the scheme, the PULL model failure detection is used as the basic failure detection strategy. Pirani *et al.* [22] proposed a failure detection algorithm based on prediction of nodes' speed. In this algorithm, the neighbor nodes can calculate the movement speed of target node, then the target node failure can be determined by the speed. The above approaches are all qualitative solutions to the problem of failure detection, and thus they cannot quantitatively control the output of failure detection.

C. QoS METRICS OF A FAILURE DETECTOR

Many distributed applications have some timing constraint on the behavior of failure detectors. It is not acceptable for a node to be suspected hours after it has crashed or for the failure detector to output several false positives. To solve this problem, Chen *et al.* [23] proposed a series of metrics to specify the QoS of a failure detector: 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 and 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 failure detector.

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.

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

Considering the impact of detection overhead on the performance of VANETs, we also use detection overhead as an important metric to describe the performance of a failure detector.

Detection overhead (O_D) is the traffic generated to detect a failure node. We can measure the detection overhead of detecting a node by the average number of detection messages generated per unit of time.

III. IMPLEMENTATION OF VC-FD

A. SYSTEM MODEL

We consider a partially synchronous system consisting of a finite set of nodes $\Pi = \{p_1,p_2,\ldots,p_n\}$. 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 failure detectors are implemented using the UDP protocol, we assume that the communication channel between nodes is a fair-lossy channel [15], i.e., no message can be copied or modified, no new message can be created, and if a node p continues sending a message m to node q, q will eventually receive m.

We assume the existence of some global time (unbeknownst 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).

B. PRINCIPLE OF VC-FD

In VANETs, the basic failure detection strategy employs the PULL model [24], [25] as the implementation of the failure detector. To simplify the description, suppose that the system consists of only two nodes p and q, where q is monitoring p. Node q sends the detection message "are you alive?" to node p every η s. After receiving the detection message, node p immediately replies with an acknowledgement message "I'm alive" to indicate its status. If node q does not receive k consecutive acknowledgement messages, it determines the status of node p to be failure; otherwise, it determines the status of node p to be correct. This failure detection strategy is the basis for further research on failure detection in VANETs.

According to the architecture of VANETs, we propose a hierarchical failure detector VC-FD, which can obtain the status of target nodes by cooperation with neighboring nodes. For VC-FD, all of the nodes in VANETs can be grouped by RSUs, and a group is composed of an RSU and several vehicular nodes. In every group, the different nodes can detect each other and share failure information of a target node with neighboring nodes and the RSU. For intergroup communication, RSUs as the leader nodes can exchange the failure information of nodes and detect other RSUs. If the RSU fails, the failure information will be broadcast to other RSUs, and the remaining nodes in the same group will be assigned to other groups according to routing distance.

Fig. 2 shows the principle of VC-FD. For nodes of the same group, failure of a target node can be learned by monitoring nodes based on their own detection messages or notification messages from other monitoring nodes. For example,

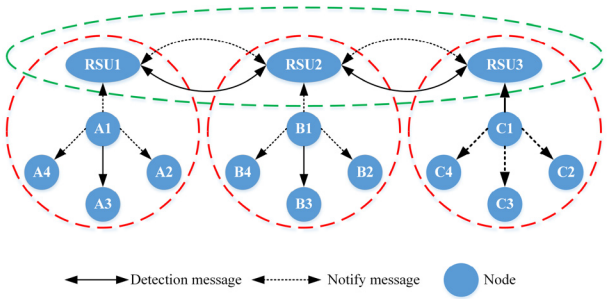

FIGURE 2. Principle of VC-FD.

the target node A3 is detected by the monitoring nodes A1, A2, and A4 in the same group. If monitoring node A1 does not receive any notification message from the other monitoring nodes (A2, A4), it will broadcast the failure information to the other monitoring nodes (A2, A4) and RSU when it finds the failure node. The monitoring nodes (A2, A4) determine the target node A3 failure after receiving the notification message from monitoring node A1 and then stop the detection process. In this manner, the detection time can be reduced due to detection-initiated asynchrony and notification messages.

For the nodes of different groups, target node failure can be determined by monitoring nodes based on notification messages exchanged by RSUs. For example, monitoring node A1 can obtain the status of target node C3 by querying the RSU of the same group rather than detecting target node C3 directly. In this manner, the influence of changes of system topology can be reduced so that the failure node can be detected in a timely manner.

Monitoring RSUs are able to determine target RSU failure based on their own detection messages or notification messages from other monitoring RSUs. For example, if the RSU fails in the same group as node B1, the RSU of another group will broadcast the failure information to other RSUs when it finds the failure RSU firstly. Then, the nodes B1, B2, B3, and B4 will be assigned to other groups according to routing distance.

C. RELATIONSHIP OF QoS METRICS AND DETECTION PARAMETERS

To quantitatively evaluate the performance of VC-FD, the function relationships are established between detection parameters and Quality of Service (QoS) metrics. The primary QoS metrics and parameters of VC-FD are given in Table 1.

Detection time as a primary QoS metric is used to evaluate the speed of failure detection. In BFD, target node failure is determined when k consecutive acknowledgement messages are not received. We assume that the failure probability of a node follows a uniform distribution on $(0, k\eta)$, so the average time it takes BFD to detect that a neighbor has failed is

$$T_D = \frac{k\eta}{2} \tag{1}$$

Based on the BFD, VC-FD also determines that target node failure has occurred when notification messages are received.

TABLE 1. Symbol and implication.

Symbol	Implication
T_D	detection time
λ_M	mistake rate
O_D	detection overhead
η	sending interval of message
d	number of neighboring nodes
k	number of message retransmissions
p_l	message loss probability
f_p	probability of node failure alone

Thus, we can obtain the detection times of two extreme cases easily, i.e., the longest detection time and shortest detection time. The longest detection time is $k\eta$ s. This is the case in which the target node fails immediately after an acknowledgement message is sent out, and VC-FD gets no notification messages from the other monitoring nodes in the same group. Without help from other nodes, VC-FD is essentially the same as the basic failure detection strategy. Hence, it can only determine that the target node has failed after it has missed k consecutive acknowledgement messages. Therefore, the longest detection time is $k\eta$ s. The shortest detection time is almost 0. This is the case when VC-FD receives a notification message from other monitoring nodes in the same group when it is just about to send a detection message to the target node. Thus, the shortest detection time is almost 0. In fact, we are interested in the average time for a notification message from at least one of the d neighbors to arrive at VC-FD. This is because the probability of receiving a notification message from neighboring nodes in the average amount of time is much higher than the probability of receiving a notification message from neighboring nodes in 0 time. According to a well-known order statistic theorem [26], the probability of VC-FD receiving at least one notification message from d uniformly distributed neighboring nodes on $(0,k\eta)$ follows the $k\eta\beta_d$ distribution, where β_d is the beta distribution with parameters 1 and d. The expected value of β_d is $1/(d+1)$. Thus, the average time for VC-FD to receive at least one notification message from neighboring nodes is $k\eta/(d+1)$. Considering the case that no neighboring node is in the group, we assume that the failure probability of a node follows a uniform distribution on $(0, \eta)$, and the average time it takes VC-FD to detect that a neighbor has failed is

$$T_D = \frac{d+2}{2(d+1)}k\eta \tag{2}$$

Mistake rate as a primary QoS metric is used to evaluate the accuracy of failure detection. In BFD, it makes a mistake when k consecutive acknowledgement messages are lost. Because BFD employs the PULL model, the messages can be lost during the sending or receiving phases. With that, the mistake rate of BFD is simply

$$\lambda_M = 2p_l^k \tag{3}$$

where p_l is the message loss probability.

In VC-FD, if no notification message is received, the mistake rate of VC-FD is the same as BFD (the mistake rate

is $2p_l^k$). If at least a notification message from an error detection of d neighboring nodes is received, VC-FD will make a mistake. The probability that all the d neighboring nodes do not make mistake is $(1-2p_l^k)^d$. Thus the probability that at least a neighboring node makes mistake is $1-(1-2p_l^k)^d$. With that, the mistake rate of VC-FD is simply

$$\lambda_M = 2p_l^k + 1 - (1-2p_l^k)^d \tag{4}$$

In BFD, the monitoring node sends a detection message to the target node and receives an acknowledgement message every η s if the target node is not in a failure state and no messages are lost in the transmission process. That means that a detection message and an acknowledgement message are generated every η s, if the target node does not fail. In fact, the probability of receiving an acknowledgement message by the monitoring node is (1-f_p)(1-p_l), where f_p is the probability of node failure alone. Thus the number of acknowledge message is (1-f_p)(1-p_l) every η s. With that, the detection overhead of BFD is

$$O_D = \frac{1}{\eta} + \frac{(1-f_p)(1-p_l)}{\eta} \tag{5}$$

For VC-FD, its detection overhead will increase due to notification messages if the target node is in a failure state. There are two situations that cause the notification message to increase. The first case is that the target node does fail (the probability is f_p). Thus the monitoring node will send d-1 notification messages to neighboring nodes every η s. The second case is that the target node is not failure, only the k consecutive acknowledgement messages are lost (the probability is $(1-f_p)p_l^k$). Thus the monitoring node will also send d-1 notification messages to neighboring nodes every η s. With that, the detection overhead of VC-FD is

$$\begin{aligned} O_D = &\frac{1}{\eta} + \frac{(1-f_p)(1-p_l)}{\eta} \\ &+ \frac{f_p(d-1)}{\eta} \\ &+ \frac{(1-f_p)}{\eta} p_l^k (d-1) \end{aligned} \tag{6}$$

D. IMPLEMENTATATION OF THE VC-FD ALGORITHM

From all of the above, VC-FD algorithm is shown to consist of a basic failure detection algorithm and cooperative mechanism. To simply the description, we select monitoring node q and detected node p as the described object. In the basic failure detection algorithm (shown as Algorithm 1), after the monitoring node sends a detection message every η s, the detected node replies with an acknowledgement message. If the monitoring node does not receive an acknowledgement message from the detected node within η s, it considers the detection message to have been lost. When a detection message loss occurs, at most k-1 detection messages will be sent. Then, monitoring node q will determine the status of node p if it does not receive any acknowledgement message.

In the VC-FD algorithm (shown as Algorithm 2), it determines the detected node failure for monitoring nodes based on their own detection messages or notification messages from other monitoring nodes. For a monitoring node, it will determine the detected node failure if it receives notification messages from other monitoring nodes. Otherwise, the monitoring node will send a detection message to the detected node every η s. It is also able to determine the detected node failure if it does not receive any acknowledgement message. Then, it sends notification messages to neighboring nodes. For the detected node, it will reply with an acknowledgement message to the monitoring node if it receives a detection message.

Algorithm 1 Basic Failure Detection (BFD)

Input: η;
Output: *suspectlist*[];
Node q: /*monitoring node*/
for all i > 0 **do**
  at time $i \cdot \eta$ send detection message mq_i to node p;
  if don't receive any acknowledge message after consecutive k detection messages **then**
    add p to *suspectlist*[];
  end if
end for
Node p: /*detected node*/
if receive mq from q **then**
  send ma to q;
end if

Algorithm 2 VC-FD

Input: η, k;
Output: *suspectlist*[];
Node q: /*monitoring node*/
if receive notify message **do**
  add p to *suspectlist*[];
else
  for all $i > 0$ **do**
    at time $i \cdot \eta$ send detection message mq_i to node p;
    if don't receive any acknowledge message after consecutive k detection messages **then**
      add p to *suspectlist*[];
      send notify messages to neighbor nodes;
    end if
  end for
end if
Node p: /*detected node*/
if receive mq from q then
  send ma to q;
end if

IV. SIMULATION RESULTS AND ANALYSIS

A. EXPERIMENTAL SETUP

To evaluate and analyze the performance of VC-FD, we design and implement the experiment based on NS2. In this experiment, we use GT-ITM [27] to generate

2000-node transit-stub topologies as the underlying network. One stub node is randomly chosen as the source, and other end-hosts are randomly distributed in stub domains. The application-level distance (path latency) between two end-hosts is the sum of link latencies on the shortest path between them. Referring to the data available on the Internet, the path latency ranges from 1 to 220 ms, with the average equal to 96 ms. The number of neighboring nodes that each node can have in the VANETs is uniformly distributed within the range $d \in [2,30]$. Note that d potentially decides the average size of the monitoring groups in the VANETs. Thus, we control the average monitoring group size by varying d. In the experiment, $d = 10$ if not mentioned otherwise.

In the experiment, the nodes join and leave the system dynamically, which is modeled as a Poisson process with a leaving and joining rate $\lambda = 0.2$/s. The experiment is carried out for two hours. We use the Gilbert model, which has often been used to reflect the bursty losses observed over the Internet, to simulate the message loss. The message loss probability is $p_l = 1\%$ if not mentioned otherwise. This relatively high message loss probability biases against the cooperative approach. In an environment with less message loss, the cooperative approach can perform better than in the following simulation. The sending interval η of detection messages is set to be 0.5 s and 1 s. In this experimental setting, the VC-FD algorithm and basic failure detection algorithm are verified and analyzed based on detection speed, accuracy and overhead.

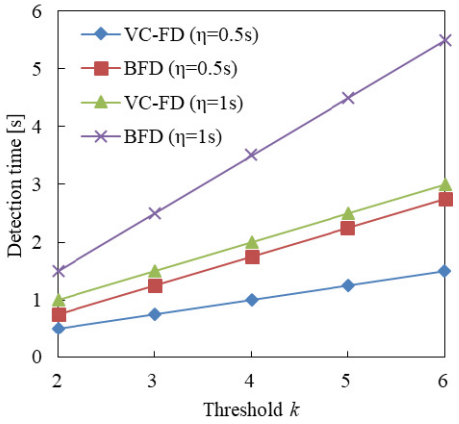

FIGURE 3. Detection time vs. threshold.

B. DETECTION TIME

Detection time as a primary QoS metric is used to evaluate the detection speed, which has an important impact on the performance of the upper application. For example, task completion time, network throughput and streaming frame loss probability can be affected by the detection time. For VC-FD, it is able to improve the detection time by sharing detection results obviously (shown as Fig. 3). Fig. 3 shows that both failure detection algorithms (VC-FD and BFD) are compared at different sending intervals ($\eta = 0.5$ s and $\eta = 1$ s). From Fig. 3, the improvement of detection time for VC-FD is more

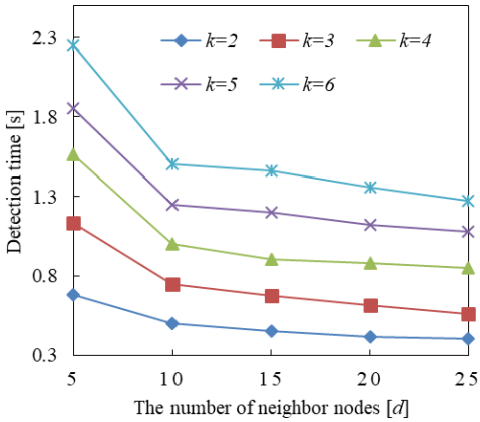

FIGURE 4. Detection time vs. number of neighboring nodes with $\eta = 0.5$ s.

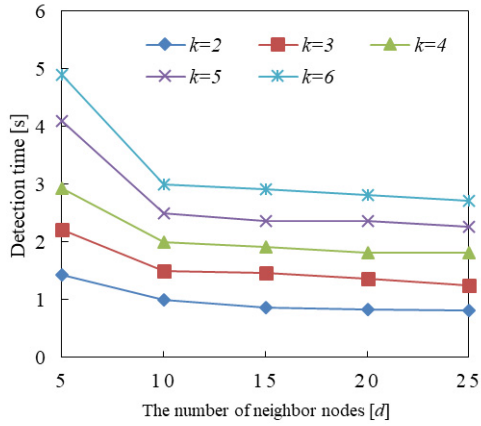

FIGURE 5. Detection time vs. number of neighboring nodes with $\eta = 1$ s.

obvious than for BFD with the increase of the value of k. When $k = 6$ and $\eta = 1$ s, the improvement of detection time for VC-FD is 45% compared to BFD. This is because a monitoring node can obtain the failure information of a detected node by receiving notification messages from other monitoring nodes. The detection periods of all the monitoring nodes are different, so some monitoring nodes can obtain the failure information of a detected node before their own detection. With that, VC-FD, which employs a detection result sharing mechanism, improves the detection time obviously.

In VC-FD, the number of neighboring nodes is also an important factor for detection time. Figs. 4-5 show the relationship between detection time and the number of neighboring nodes. From the figures, it is shown that the detection time will decrease with the increase of the number of neighboring nodes. However, the curve becomes less steep. This may be because the detection time is affected by nodes joining and leaving dynamically.

C. MISTAKE RATE

Mistake rate is also a primary QoS metric used to evaluate the detection accuracy. Fig. 6 shows a comparison of mistake rate for both VC-FD and BFD. From the figure, the mistake rate will decrease with the increase of the threshold k. This is because the mistakes that are made by the loss of heartbeat can be improved by retransmitting heartbeats. When the

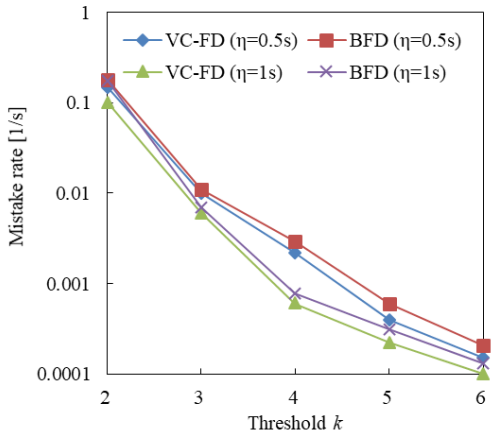

FIGURE 6. Mistake rate vs. threshold.

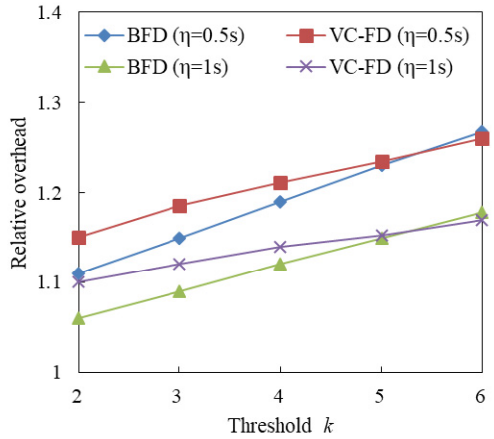

FIGURE 7. Relative overhead vs. threshold.

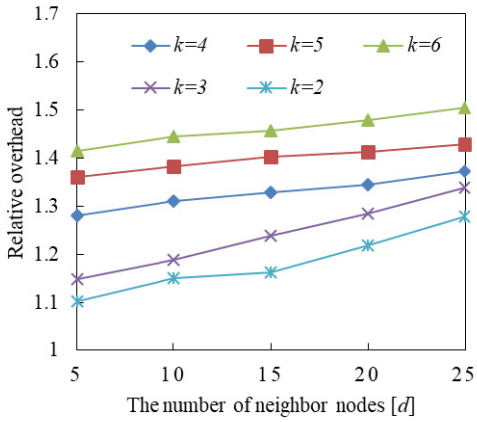

FIGURE 8. Relative overhead vs. number of neighboring nodes with $\eta = 0.5$ s.

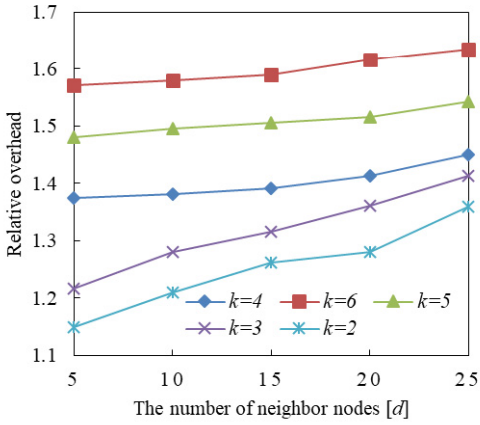

FIGURE 9. Relative overhead vs. number of neighboring nodes with $\eta = 1$ s.

threshold is equal, the mistake rate of VC-FD is higher than that of BFD. VC-FD can determine the detected node failure based on itself or other monitoring nodes. This method avoids the effect of communication link failure and reduces the error detection to improve the mistake rate of VC-FD. When the threshold k becomes larger, the mistake rate of VC-FD has an improvement of up to 25% compared to BFD. In addition, when $k = 6$ and $\eta = 0.5$, the average time between error detection is 6667 s in VC-FD. This can meet the requirements of most users or applications.

D. DETECTION OVERHEAD

To evaluate the overhead of various detection approaches, we measure the number of messages sent per second for detection purposes. For BFD, we only need to record the number of detection messages. For the VC-FD, we record not only the number of detection messages but also the number of notification messages. To simply the description, we define the relative overhead to compare VC-FD with BFD. The relative overhead is the ratio of actual overhead between VC-FD and BFD under the same experimental configuration.

Fig. 7 depicts the relative overhead of the two failure approaches with different sending intervals η. From the figure, the relative overhead becomes higher with the increase of the threshold k. This is because the number of detection messages increase in every detection period with the increase of the threshold k. When $k < 5$, the relative overhead of VC-FD is higher than that of BFD. This is because more notification messages are sent by VC-FD, especially when the threshold k is small. When $k >= 5$, the relative overhead of VC-FD is lower than that of BFD. This is because the accurate notification messages can help to decrease the number of detection messages in VC-FD.

Figs. 8-9 depict the relationship between the relative overhead and number of neighboring nodes in VC-FD. From the figures, the relative overhead becomes higher with the increase of the number of neighboring nodes, regardless of whether the sending interval is big or small. This may be because more neighboring nodes consume more detection messages and notification messages. By the way, the threshold k is also an important factor that affects the overhead in VC-FD. The relative overhead becomes higher with the increase of the threshold k.

V. CONCLUSION

Failure detection plays a very important role in VANETs. In this paper, we have introduced a hierarchical failure detector VC-FD based on the architecture of VANETs. By using the RSU to group the vehicles and adopting the detection result sharing mechanism, VC-FD can deal with the high

mobility of vehicles to improve the detection accuracy and speed. Meanwhile, we have built the function relationship between detection parameters and QoS metrics so that quantitative output of QoS can be realized.

We have built an experimental platform for performance evaluation of our proposed algorithm using NS2 and GT-ITM. The results of the experiments have shown that our proposed algorithm can significantly outperform the basic failure detection algorithm of VANETs with the threshold k set to (2,6) and two heartbeat-sending intervals (0.5 s and 1 s). The detection time is improved at most 45%, and the mistake rate is also improved at most 25% with the similar overhead. In future work, we plan to optimize our algorithm to further improve overhead considering other factors, for example, the behavior of vehicles.

REFERENCES

[1] E. Cambruzzi, J. M. Farines, R. J. Macedo, and W. Kraus, "An adaptive failure detection system for vehicular ad-hoc networks," in *Proc. IEEE Intell. Vehicles Symp.*, San Diego, CA, USA, Jun. 2010, pp. 603–608.
[2] E. Evdokimova, A. Vinel, N. Lyamin, and D. Fiems, "Internet provisioning in VANETs: Performance modeling of drive-thru scenarios," *IEEE Trans. Intell. Transp. Syst.*, to be published.
[3] S. Bourebia, H. Laghmara, B. Hilt, F. Drouhin, S. Bindel, J. Ledy, J.-P. Lauffenburger and P. Lorenz, "A belief function-based forecasting link breakage indicator for VANETs," *Wireless Netw.*, vol. 25, pp. 1–16, Mar. 2019.
[4] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R. A. F. Mini, and A. A. F. Loureiro, "Data communication in VANETs: Protocols, applications and challenges," *Ad Hoc Netw.*, vol. 44, no. 1, pp. 90–103, Jul. 2016.
[5] F. Jamil, A. Javaid, T. Umer, and M. H. Rehmani, "A comprehensive survey of network coding in vehicular ad-hoc networks," *Wireless Netw.*, vol. 23, no. 8, pp. 2395–2414, Nov. 2017.
[6] L. A. Villas, A. Boukerche, G. Maia, R. W. Pazzi, and A. A. Loureiro, "DRIVE: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks," *Comput. Netw.*, vol. 75, no. 1, pp. 381–394, Dec. 2014.
[7] N. Aljeri, M. Almulla, and A. Boukerche, "An efficient fault detection and diagnosis protocolfor vehicular networks," in *Proc. ACM 3rd Int. Symp. Design Anal. Intell. Veh. Netw. Appl.*, Barcelona, Spain, 2013, pp. 23–29.
[8] N. Xiong, A. V. Vasilakos, J. Wu, Y. R. Yang, A. Rindos, Y. Zhou, W.-Z. Song, and Y. Pan, "A self-tuning failure detection scheme for cloud computing service," in *Proc. IPDPS*, Shanghai, China, May 2012, pp. 668–679.
[9] A. Tomsic, P. Sens, J. Garcia, L. Arantes, and J. Sopena, "2W-FD: A failure detector algorithm with QoS," in *Proc. IPDPS*, Hyderabad, India, May 2015, pp. 885–893.
[10] N. Hayashibara, X. Defago, R. Yard, and T. Katayama, "The φ accrual failure detector," in *Proc. SRDS*, Florianpolis, Brazil, Oct. 2004, pp. 66–78.
[11] G. Bosilca, A. Bouteiller, A. Guermouche, T. Herault, Y. Robert, P. Sens, and J. Dongarra, "A failure detector for HPC platforms," *Int. J. High Perform. Comput. Appl.*, vol. 32, no. 1, pp. 139–158, Jul. 2018.
[12] Y. He, X. Jiang, C. Dai, and Z. Fan, "Self-adaptive failure detector for peer-to-peer distributed system considering the link faults," in *Proc. APPT*, Santiago de Compostela, Spain, 2017, pp. 64–75.
[13] R. C. Turchetti, E. P. Duarte, Jr., L. Arantes, and P. Sens, "A QoS-configurable failure detection service for Internet applications," *J. Internet Services Appl.*, vol. 7, no. 9, Apr. 2016, Art. no. 9.
[14] O. Senouci, Z. Aliouat, and S. Harous, "MCA-V2I: A multi-hop clustering approach over vehicle-to-Internet communication for improving VANETs performances," *Future Gener. Comput. Syst.*, vol. 96, no. 1, pp. 309–323, Jul. 2019.
[15] P. Felber, X. Defago, R. Guerraoui, and P. Oser, "Failure detectors as first class objects," in *Proc. DOA*, Edinburgh, U.K., Sep. 1999, pp. 132–141.
[16] M. Bertier, O. Marin, and P. Sens, "Performance analysis of a hierarchical failure detector," in *Proc. DSN*, San Francisco, CA, USA, 2003, pp. 635–644.
[17] Y. Yang, J. Li, and S. Song, "A double-layer failure detection algorithm based on weight," in *Proc. ICCT*, Chengdu, China, Nov. 2012, pp. 904–908.
[18] N. Apolónia, F. Freitag, L. Navarro, S. Girdzijauskas, and V. Vlassov, "Gossip-based service monitoring platform for wireless edge cloud computing," in *Proc. ICNSC*, Calabria, Italy, May 2017, pp. 789–794.
[19] S. Q. Zhuang, D. Geel, I. Stoica, and R. H. Katz, "On failure detection algorithms in overlay networks," in *Proc. INFOCOM*, Miami, FL, USA, Mar. 2005, pp. 2112–2123.
[20] E. M. Salvador, D. F. Macedo, and J. M. S. Nogueira, "HE-MAN: Hierarchical management for vehicular delay-tolerant networks," *J Netw. Syst. Manage.*, vol. 26, no. 3, pp. 663–685, Jul. 2018.
[21] K. Khatkar, N. Batra, and R. P. Singh, "Fault tolerance approach for improving connectivity in vehicular ad hoc network," *Int. J. Appl. Eng. Res.*, vol. 13, no. 2, pp. 823–829, Jun. 2018.
[22] M. Pirani, E. Hashemi, A. Khajepour, B. Fidan, B. Litkouhi, S.-K. Chen, and S. Sundaram, "Cooperative vehicle speed fault diagnosis and correction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 783–789, Feb. 2018.
[23] W. Chen, S. Toueg, and M. K. Aguilera, "On the quality of service of failure detectors," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 561–580, May 2002.
[24] K. Abrougui, A. Boukerche, and H. Ramadan, "Performance evaluation of an efficient fault tolerant service discovery protocol for vehicular networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1424–1435, Sep. 2012.
[25] M. S. Kakkasageri and S. S. Manvi, "Information management in vehicular ad hoc networks: A review," *J. Netw. Comput. Appl.*, vol. 39, no. 1, pp. 334–350, Mar. 2014.
[26] R. Durrett, *Probability: Theory and Examples*. London, U.K.: Cambridge Univ. Press, vol. 2019, pp. 83–90.
[27] *GT-ITM: Georgia Tech Internetwork Topology Models*. Accessed: 2018. [online]. Available: https://www.cc.gatech.edu/projects/gtitm

JIAXI LIU received the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, in 2019. He is currently a Lecturer with the Nanjing University of Posts and Telecommunication. His research interests include fault tolerant computers, failure detection in distributed systems, and mobile computing.

FEI DING received the Ph.D. degree in instrument science and technology from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, in 2010, and received the Ph.D. degree from the School of Information Science and Engineering, Southeast University. He was an Internet of Things (IoT) Research Leader with the Research and Development Center, China Mobile Group Jiangsu Company Ltd., Nanjing, and also was a Postdoctoral Researcher with the School of Information Science and Engineering, Southeast University. He is currently an Associate Professor with the School of IoT, Nanjing University of Posts and Telecommunications, Nanjing. His research interests include wireless sensor networks, the IoT, and mobile communication related key technologies.

DENGYIN ZHANG received the B.S., M.S., and Ph.D. degrees from the Nanjing University of Posts and Telecommunication, Nanjing, China, in 1986, 1989, and 2004, respectively. He was with the Digital Media Lab, Umea University, Sweden, as a Visiting Scholar, from 2007 to 2008. He is currently a Professor and Department Manager of the School of Internet of Things, Nanjing University of Posts and Telecommunication. His research interests include signal and information processing, networking technique, and information security.