Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6833535, 16 pages
https://doi.org/10.1155/2022/6833535

Research Article

A Partitioned DAG Distributed Ledger with Local Consistency for Vehicular Reputation Management

Naipeng Li,[1] **Yuchun Guo**,[1] **Yishuai Chen**,[1] **and Jinchuan Chai**[2]

[1]*School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2]*National Railway Track Test Center, China Academy of Railway Sciences, Beijing 100015, China*

Correspondence should be addressed to Yishuai Chen; yschen@bjtu.edu.cn

Received 11 November 2021; Revised 3 March 2022; Accepted 8 March 2022; Published 23 March 2022

Academic Editor: Qingqi Pei

Vehicular reputation maintenance with distributed ledger is aimed at establishing trust among vehicles randomly meeting in a Vehicular Ad-hoc Network (VANET). It is, however, challenging in VANET, as congested areas in road networks, brought by traffic tides or accidents, challenge the ledger performance. Meanwhile, the reputation update is highly dependent on transaction consensus of the distributed ledger. To solve the problem, this paper proposes deploying directed acyclic graph- (DAG-) based distributed ledgers on vehicles, which use the vehicular distribution to adapt the unpredictable reputation update. Specifically, we first propose a partitioned DAG-based distributed ledger to manage vehicular reputation in partitioned VANET. Secondly, we introduce a novel reputation evaluation method to encourage vehicles to contribute to VANET interaction and ledger consensus maintenance, which can remedy the topology churn of the ledger network due to the mobility of VANET. Finally, we design a reputation update method based on the consistency of transactions in the partition to facilitate trust establishment. Experimental results on a real-world dataset show that the proposed ledger and reputation update method is effective and feasible in the large-scale dynamic VANET.

1. Introduction

With the rapid evolution of Vehicular Ad-hoc Networks (VANETs) and intelligent technology, intelligent vehicles have further demands for exchanging information with surrounding smart objects like other intelligent vehicles, smart traffic lights, and Road Side Units (RSUs) [1, 2]. However, the unique features of VANETs, such as high mobility and volatility, make the antiattack and privacy protection become major concerns [17]. Establishing the trust for received information or connectable nodes needs a vehicular reputation management system. Recently, due to providing privacy protection and decentralized trust among unfamiliar vehicles [17], employing Distributed Ledger Technologies (DLTs), such as blockchain, for vehicular reputation management has become a hot research topic [1, 4, 6, 9].

Nevertheless, a vehicular reputation management system with distributed ledger also faces two challenges. The first challenge is ledger maintenance. To maintain the consistency of the transaction, the distributed ledger needs consensus mechanisms like Proof of Work (PoW) [7] or Proof of Stake (PoS) [8] in the blockchain ledger. However, it is challenging for the VANET node to satisfy the requirements of consensus mechanisms, e.g., computing power and stable communication route [6]. The second challenge is trust establishment. Evaluating the other's reputation is an excellent way to establish trust for an interaction. A node can evaluate reputation for one time when the interaction begins rather than assessing the context of each exchanging message. However, the VANET, a decentralized network, challenges the reputation update.

Existing studies for vehicular reputation management are mainly based on the blockchain platform [6, 20]. There are

two kinds of nodes in a blockchain platform, the full node and the simplified node. The former keeps the entire blockchain ledger and composes the backbone blockchain network. The latter only creates and consumes transactions that store reputation ratings on others or some nodes' historical behaviors but do not maintain the ledger. The researchers use the RSUs as the full nodes to satisfy computational and stable bandwidth resources and the vehicles as the simplified nodes to quantify and encapsulate the vehicular interactive behaviors into the transactions [3, 5, 6, 9, 17, 20]. However, it is tricky for an RSU-based blockchain ledger to guarantee timely reputation updates. Vehicles can only connect and issue transactions to nearby vehicles or RSUs [21]. When a traffic jam or accident occurs, the vehicle density will dramatically increase in the spot and overload adjacent RSUs, which delays the consensus of the ledger and update of reputation.

Distributed ledger with a directed acyclic graph- (DAG-) based architecture seems to have better scalability than traditional blockchain architectures, which provide promising solutions to solve the issue of uneven reputation update workload. Firstly, the DAG-based distributed ledger (DDL) has a higher throughput than the blockchain. Instead of competing for the block-level consensus, DDL verifies and approves the transactions in different parts of the ledger network parallelly with homogenous nodes. A DDL node is required to approve recently issued transactions to help these pending transactions receive enough approvals quickly. Secondly, some DDL systems, such as the Tangle [10], are designed for the IoT scenario composed of low-resource nodes, e.g., vehicles. Although some works have been done on DDL-enabled VANET, they mainly carried out the feasibility tests [13, 14], and two key issues have still not been properly solved. (1) DDL needs to sort the parallelly approved transactions [11], but this brings about the complexity of the design. (2) DDL requires enough nodes to ensure the high throughput and security of the ledger. It is a challenge to guarantee this in the mobile VANET.

To address the above challenges, this paper proposes a partitioned DDL with local consistency for reputation management in VANETs. We designed the ledger and reputation update method based on the insight of spatiotemporal sensitivity. On the one hand, limited by the sensing range, some interactions occur only between the nearby vehicles [12]. For example, the traffic lights at an intersection are only helpful for nearby vehicles that also only these vehicles can check the trustworthiness of light information instantly. On the other hand, unlike financial applications, reputation management in VANETs does not require a transaction to reach a consensus among all nodes. Specifically, a vehicle needs someone's reputation only when establishing trust with a meeting vehicle, so the vehicles with different routes do not need each other's reputation in practice. We argue that a vehicle could independently choose which vehicles to follow according to its own itinerary needs. Ensuring the related transactions are consistent among the vehicles in a particular range is enough. The main contributions of this paper include the following.

(i) We design a partitioned DAG-based distributed ledger based on the Tangle architecture for reputation management in the VANETs

(ii) We present a vehicular reputation evaluation method by assessing the node's interactive quality and the contribution to maintaining the transaction consensus

(iii) We propose a reputation update method based on local transactional consistency to reduce the update latency and improve the trust establishment

(iv) To demonstrate the effectiveness of the proposed ledger framework, we conduct the simulations on a real-world dataset, and simulation results reveal that the proposed framework is effective and feasible in the large-scale VANETs and the reputation update delay also converges when the VANETs size is growing exponentially

The remainder of this paper is organized as follows. Section 2 surveys the existing reputation system for VANETs and summarises the related DDL works. Section 3 describes the framework overview and system model. The details of the proposed ledger are carried out in Section 4. Section 5 proposes the reputation definition and update method. We conduct simulations and discuss the numeric results in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

In this section, we classify existing DLT-based reputation/trust management systems and present existing works about expanding the throughput of the distributed ledger systems.

2.1. DLT-Enabled Reputation/Trust Management in VANET. In a DLT-enabled reputation/trust management system in VANET, the distributed ledger helps the vehicles build the consensus of the data, trust, or opinion related to the participant's reputation. The state-of-the-art systems can be classified into two categories: access-to-trust system and evaluate-to-trust system. We introduce them in detail as follows:

(1) Access-to-trust system

Access-to-trust systems require that any nodes get permission first before they are considered trustworthy and build trusted communication [15, 16]. In general, the systems maintain a white list or black list to control this communication permission. Lu et al. [17] utilize two blockchains to record the workflow of the Certificate Authority (CA) and management history of all vehicles separately, and the former monitors the credibility of the CA, and the latter maintains the reputation of the system nodes and assists the CA in the issuance of certificates. With the development of smart contract [18], Javaid et al. [4] and Liu et al. [19] all adopted the smart contract to control the registering and access of the honest vehicles, and only the vehicles with permission can communicate with each other freely. Furthermore, Wang et al. [20] use smart contracts to manage the access of vehicles, and vehicles can obtain the evaluation results of the other vehicle's reputation by submitting the request to the specific smart contract. To overcome the dynamic network size of the VANET, Javaid et al. [4] modified the Proof of Work (PoW) mechanism to

adapt to the incoming traffic generated by the vehicles. Kudva et al. [22], Khalid et al. [23], and Liu et al. [19] build their systems based on the consortium blockchain platform, which operates only with a fixed number of preauthorized nodes so it can assign to some powerful nodes to keep the performance of their ledger system. All of the above works focus on communication efficiency but lack investigation into the incentives of the nodes in a decentralized system.

(2) Evaluate-to-trust system

Evaluate-to-trust systems directly assess the credibility of the transmitted data, including based on the voting of different data sources about the same event, the reputation of the source sender, or even the empirical probability of the event occurrence. Kang et al. [24]use the interaction frequency, event timelines, and trajectory similarity of the source vehicles to evaluate their message's credibility; Road Side Units (RSUs) collect the reputation opinions or other shared data and ensure the consensus of these records by PoW. To solve similar problems, Yang et al. [6] use the location of the sending vehicles to evaluate the message's credibility, and RSUs collect multiple messages that report the same event from different vehicles, calculate the sending vehicles' offsets, and add them to the blockchain through a consensus mechanism combining PoW and Proof of Stake (PoS). Based on the above work, Lu et al. [1] required the vehicles, in addition, to continue to collect the opinions on the event after receiving it from its initiator and also to query the initiator's reputation from the RSUs to compute the event's credibility, which will eventually be transmitted back to the RSUs to update the initiator's reputation. In these systems, the evaluation of the reputation is conducted after the nodes share messages. Unlike the above systems, Li et al. [25] proposed an active detection method based on the probe to find the possible misbehavior nodes before the sharing, and the system divided the VANET into multiple fixed partitions and set some fixed powerful servers to provide stable blockchain services. However, although all the above works allow the vehicles to self-assess the message's credibility or vehicle's trustworthiness for specific interaction, they still rely on the RSUs to provide and update the vehicles' reputation.

The most existing DLT-based reputation/trust management systems adopted the blockchain platform as their infrastructure framework. They implemented the RSUs as the miner (running the backbone network of the blockchain) because they have better computing resources and more stable network links than the vehicles in VANET. However, it is easy to cause the delay of reputation update due to the limitation of the RSU bandwidth and block size [6] when the inflow traffic increases. Diallo et al. [26] and Zhang et al. [27] have tried to reduce the update delay by using other consensus technologies, such as Practical Byzantine Fault Tolerance (PBFT). However, PBFT has a high communication complexity, and the consensus cluster should not be too large. Therefore, there are no suitable flexible frameworks that can simultaneously cope with the dynamic network topology, fragility network connection, and the lower update delay requirements in VAENT.

2.2. The DAG-Based Distributed Ledger. DAG ledgers potentially offer many advantages over traditional blockchain architectures for DLTs, including scalability and faster transaction speeds (Ferraro). Many DAG-based DLT projects, such as NANO [28], Byteball [29], and Tangle [10], have been in operation for many years and have been tested in practical applications. DAG ledgers organize transactions according to the DAG structure instead of packing them into the block. The consensus is conducted in parallel and runs over the transaction level with stochastic attachment mechanisms instead of constructing a chain of blocks. All transactions must be strictly ordered by their timestamp, which forces the above methods to adopt an additional puissant and centralized component, such as the coordinator in the Tangle, to check and determine the order of all transactions. Bartolomeu et al. [14] deploy the Tangle in VANET, and their experiment result shows that the transaction confirmation delay has been significantly reduced than the blockchain solution, and the performance is comparable with Tangle's main network. However, they only validated the feasibility of DAG-based consensus deployment on VANET and there is no further research on specific VANET applications. In terms of reputation management in VANET, Zhang et al. [27] and Kang et al. [24] both try to apply the DAG structure in subregions that a miner covered to improve transaction processing speed in subregions, but their ledger is still based on the blockchain platform.

Therefore, this paper designs a DAG-based vehicular distributed ledger and implements it with the Tangle architecture to optimize reputation management in VANET.

3. Application Overview

In this section, we first present the application scenarios of sharing in VANET and the interactive model and then introduce fundamental concepts of the Tangle project as backdrop. For the clarity of the following discussion, the key notations are summarized in Table 1.

3.1. Application Model. Figure 1 shows the overview of our reputation maintenance framework where the VANET entity, including vehicles and RSUs, arises sharing interaction and the vehicular reputation, maintained by a DDL. The framework contains two layers: the sharing and vehicular reputation management layers. We require a node's reputation to be calculated by auditing the node's history interactive behavior that accumulates in the sharing layer, and the node's historical interaction is packaged into the transaction. A node will generate a transaction based on its last interaction and attach it to the DDL in the reputation management layer. To ensure all the nodes can run the DDL equally, we assume that every node has an essential computational resource and can hold a full copy of the ledger.

Another important assumption is that the peer-to-peer interactive application in VANET scenario, such as environmental awareness, is mostly geographically independent. Take the traffic density perception at an intersection as an example; Figure 1 shows that the nodes around the intersection S_1 can be seen as a subpartition of VANET, and S_1

Table 1: Notions.

Symbol	Definition	Symbol	Definition
p_i	A node that can interact with others and issue transaction	μ	The reputation
$p_{j\longrightarrow i}$	The node answered the request of p_i	σ	The metric of interactive quality
$y_{j\longrightarrow i}$	The received originally rating of $p_{j\longrightarrow i}$	η	The metric of consensus contribution
ψ_i	The transaction	R^+, R^-, R^*	The calculated ratings of positive interaction, negative interaction, and consensus contribution
$\psi_{j\longrightarrow i}$	The transaction directly approved by ψ_j	w_i	The weight of ψ_i
$\psi_{j\longrightarrow m\longrightarrow i}$	The transaction indirectly approved by ψ_j	N_c	The set of all the active nodes in a period, $\lvert N_c\rvert$ is the number of these nodes
$\{\psi\}^i$	The set of transactions that contains the historical interactions of p_i	$\mathscr{A}_i$	The cumulative weight of ψ_i
$\mathscr{L}(t)$	The set of tips at time t	Θ	The threshold of local consistency

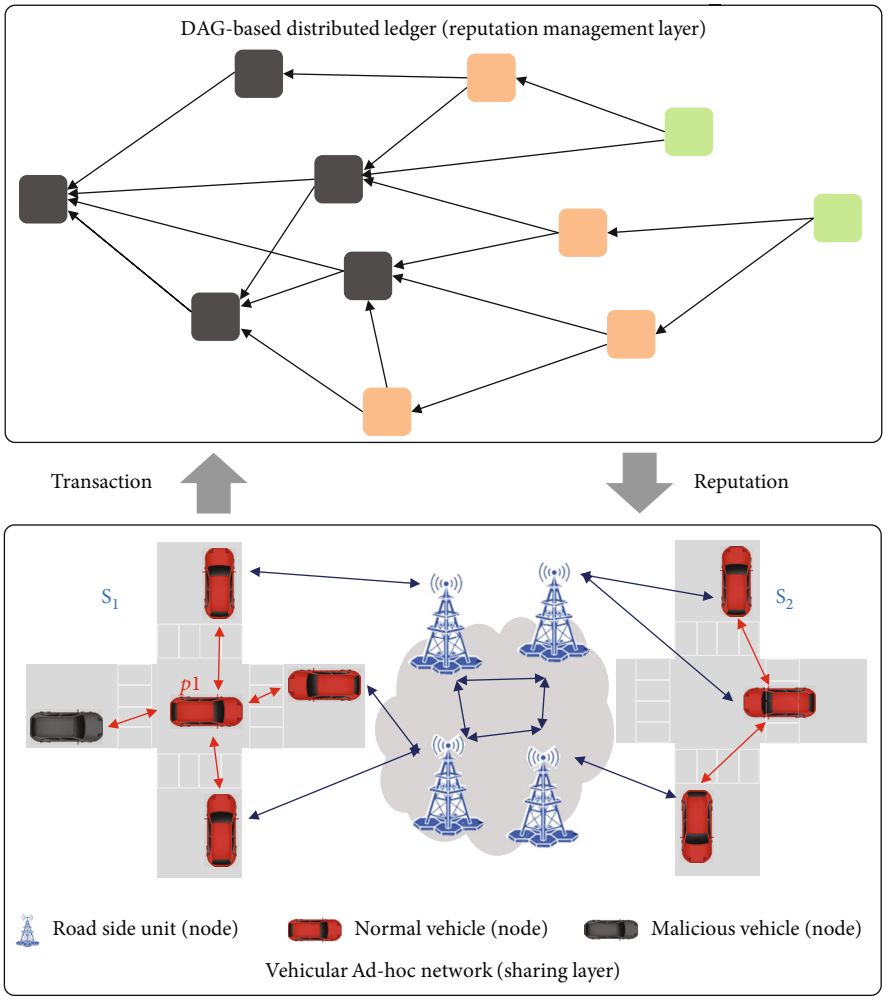


Figure 1: Overview of trust maintenance with a DAG-based distributed ledger.

consists of at least $n \geq 5$ vehicles and $m \geq 2$ RSUs. It is acceptable that all the traffic data contained in the message perceived by the p_1 can only be verified by the nodes in the same partition (that is, S_1). So, we can have $S = \{S_1, S_2, \cdots, S_K\}$, where S is the entire VANET and consists of K partitions. These partitions are connected by the RSUs, and any node belongs to at least one partition. Obviously, some applications need the nodes in different partitions to

cooperate, such as route plans, and we will discuss these more complicated scenarios in future work.

3.2. Interactive Model. In terms of most applications in VANET, we consider two typical interactive models, sharing and cooperating.

3.2.1. Sharing Model. To inform some warnings or share some knowledge, some vehicles may positively broadcast unencrypted information. Figure 2 shows that there are n nodes ($n \geq 2$, Figure 2 gives V_1 and V_2 as an example) that broadcast the knowledge of a specific event E at the same time. If a node has interests in E, it can collect a message set $\mathrm{Msg} = \{\mathrm{msg}_1, \mathrm{msg}_2, \cdots, \mathrm{msg}_n\}$. Then, the nodes need to assess the credibility of each of the received messages and evaluate a possible result of event E (because they do not know the truth about it). Based on the calculated result of event E, the node can judge whether the received messages are the same or contrary to the calculated result of E and evaluate a rating about interactive behavior of the information source node.

3.2.2. Cooperating Model. To get advice and assistance, some vehicles may ask for help from others. Figure 3 shows a general process, a node (p_1) in need broadcasts its request to the surrounding nodes first, and then, others may answer the request at their will. If there are n answered nodes, p_1 can receive some responses; for easy understanding, we also use an answering message set $\mathrm{Msg} = \{\mathrm{msg}_1, \mathrm{msg}_2, \cdots, \mathrm{msg}_n\}$ for the unified presentation with the same as sharing mode. The difference is only the p_1 can judge whether the answered node provided an effective result and evaluate the rating of interactive behavior for each answered node. To simplify the discussion, we assume that all requested nodes are honest.

3.3. DAG-Based Distributed Ledger Model. All the transactions are stored in a DAG architecture and are represented by vertexes (dark grey rectangles shown in Figure 4). A new transaction needs to approve several old (especially issued recently) transactions, and the approval relationships are represented by edges. A new edge (from the new vertex to the old vertices that stand for the selected previous transactions) is added while a new transaction is issued simultaneously. This adding process is also called transaction attachment. Some key concepts of the DDL are introduced as follows.

Transaction Approval: as shown in Figure 4, there are two types of approval relationship, direct and indirect approval. Direct approval is represented by a directed edge and indirect approval is represented by a path that consists of several transactions and direct edges connecting them. For example, E_1 indicates that ψ_c approved ψ_a, E_2 indicates ψ_b approved ψ_c, and E_4 and E_5 indicate ψ_d and ψ_e approved ψ_a indirectly.

Tip: transaction that has received no approval is called tip. A tip may be a newly issued transaction (e.g., ψ_e or ψ_f) or an old transaction but has not been approved even once (e.g., ψ_c). Define $\mathscr{L}(t)$ to be the set of tips at time t. In general, an issuing node is suggested to select tips from $\mathscr{L}(t)$ to approve, so the size of $\mathscr{L}(\mathrm{t})$ determines the growth and health of the DDL. Once a transaction is approved, it is no longer a tip, but it also needs to accumulate enough direct and indirect approval to be regarded as secure and final confirmation.

Cumulative weight: cumulative weight (CW) is a metric for measuring how trustworthy a transaction is for security consideration. Suppose $\mathscr{A}_i$ presents the CW of ψ_i and is calculated by the weights of all the transactions, including directly and indirectly, that approved ψ_i. In general, when a transaction's CW reaches (only monotonically increasing) a threshold, we say that it is confirmed, which also means it is correct and immutable. We will introduce the details of how the weights increase and threshold setting in Section 5.

4. Partitioned DAG-Based Distributed Ledger in VANET

This section first introduces how to record the details of the interaction into a transaction and how to verify. Then, we present the definition of the CW considered under the partitioned DDL. To deal with the fragility of connection and topology in VANET, we introduce a local consistency threshold and an extended tip selection algorithm to improve the throughput of transactional consensus while ensuring the ledger's security.

4.1. Historic Interaction. We need a way for the node to obtain others' reputations when establishing the trust. The existing works usually update the "balance" or "bias" of reputation. However, these solutions do not allow nodes to adjust reputations according to different situations. We consider the "auditing" method, which records the interaction details into transactions, and nodes calculate the reputation for anyone in their desired ways when needed.

Two interactive models have the different roles of the node to record each interactive detail. For a sharing model, RSUs can generate the transaction to record an interactive event in the partition it is deployed. If there are many RSUs, a rotation method can balance the workload of transaction generation. For a cooperating model, the requested node is responsible for generating the transaction when finishing a round of interaction. Take the cooperating model as the example, and we define the transaction as shown in Figure 5.

$$\Psi \coloneqq \langle d, \gamma, s, t \rangle, \tag{1}$$

where d is the interaction data, γ is the transaction approval data, and s and t represent the encrypted script and timestamp, respectively. The detail of d is

$$d \coloneqq \left\langle p_i, \left\{ p_{j \longrightarrow i} \right\}, \left\{ y_{j \longrightarrow i} \right\} \right\rangle, \tag{2}$$

where p_i denotes the issuing node that is also a requested node; $\{p_{j \longrightarrow i}\}$ and $\{y_{j \longrightarrow i}\}$ refers to all the nodes that answered the p_i and the corresponding ratings, respectively. We use a Bayesian method [6] to inference the $\{y_{j \longrightarrow i}\}$. γ represents the transaction approval relationship and composed of the following:

$$\gamma \coloneqq \langle \{\psi_{i \longrightarrow m}\}, \{\ \text{PoW Nonce}\ \}, \text{PoW target} \rangle, \tag{3}$$

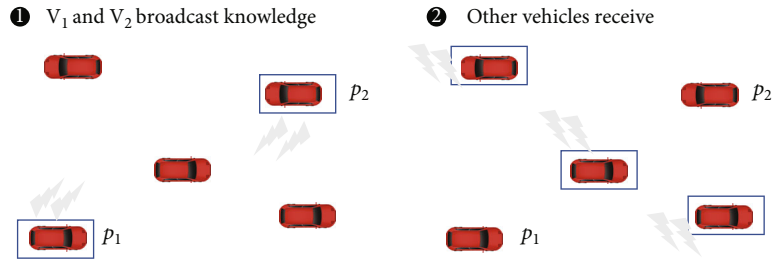


Figure 2: Two vehicles sharing data with others.

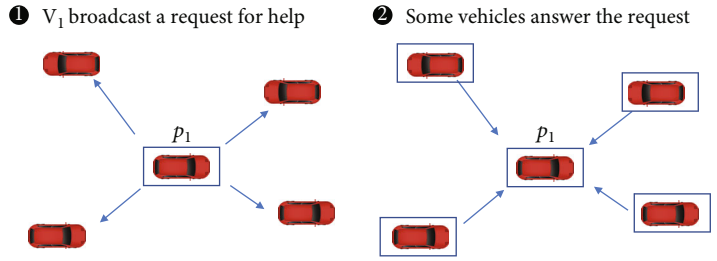


Figure 3: A vehicle asks and establishes cooperation to others.

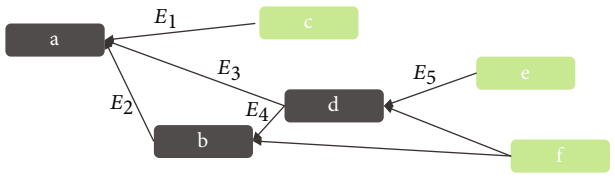


Figure 4: Representation of approval between the transactions.

where $\{\psi_{i\longrightarrow m}\}$ denotes the selected and approved transaction set. PoW Nonce and PoW target are parameters for a PoW mechanism that used to prevent malicious nodes from issuing large-scale false transactions to attack the ledger network. In general, the target is set to a small value and does not bring heavy PoW workload for a vehicle.

4.2. Transaction Verification. Each node needs to verify the newly received transaction to avoid malicious and fake transaction attacks. Moreover, the transaction weight is also calculated if it passes the verification.

The verification includes three steps; take ψ_i as an example; they are as follows:

(1) In d, whether the p_i and $\{p_{j\longrightarrow i}\}$ exist

(2) In γ, whether all the selected transactions in $\{\psi_{i\longrightarrow m}\}$ exist, and verify their PoW NONCE

(3) If step 1 and step 2 pass, calculate the transaction's weight

For step 1, we assume that all the interactive messages can be recorded with some methods, such as the smart contract among the reference nodes.

For step 2, the validation of γ requires that the selected and approved transactions must have been transmitted to all other nodes (at least in several nearby partitions) before they can be approved. In fact, the most recent tips that the issuing node cab select at time t can only be issued at $t-h$, where h, called the waiting period [10], includes both the PoW time and the minimum transmission time for transaction transmission to most nearby nodes.

For step 3, any node that received the transaction needs to calculate its weight. The weight is calculated only once and stored only at the local.

In our framework, we define the issuing node's interactive behaviors as the weight of its issuing transaction. Some nodes invest a lot, including frequently and actively responding or issuing transactions (verifying and approving the other transactions to assist the DDL). For a peer-to-peer data sharing system, it is obvious that the nodes working hard and getting higher interaction ratings should have more credibility. So, we define the weight of the transaction issued by p_i as

$$w_i = \epsilon_1 e_i + \epsilon_2 \sum_{k\in\{\psi\}^i} y_k, \tag{4}$$

where e_i denotes the number of valid transactions issued by p_i and $\{\psi\}^i$is a set of transactions that contain the historical interactions of p_i. y_k refers to the corresponding rating and $\epsilon_1 + \epsilon_2 = 1$. Obviously, e_i and the size of $\{\psi\}^i$ are changed over time, so we only calculate the weight at once when it is issued.

4.3. Cumulative Weight. The CW of a transaction can be calculated by the sum of the weights from all its successor transactions. If we set $w = 1$ for each transaction, the CW represents how much approval this transaction achieved.

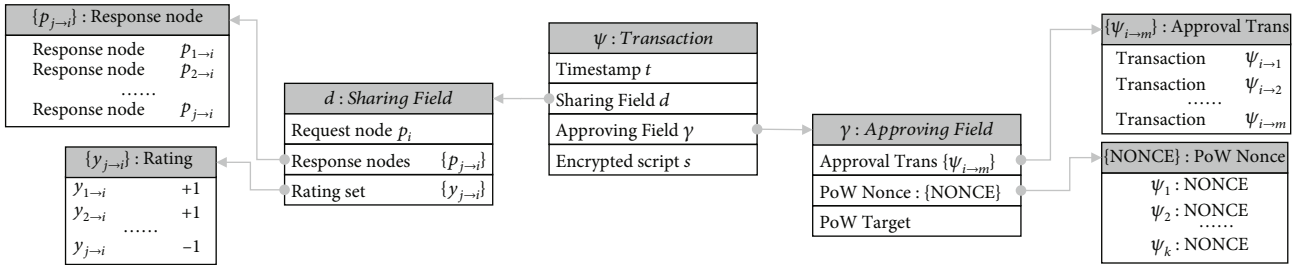


Figure 5: Data structure of the transaction.

For ψ_i, its CW is

$$\mathscr{A}_i = \sum_{j\in\{\psi_{j\longrightarrow i},\psi_{j\longrightarrow m\longrightarrow i}\}} w_j, \tag{5}$$

where $\{\psi_{j\longrightarrow i},\psi_{j\longrightarrow m\longrightarrow i}\} = \{\psi_{j\longrightarrow i}\} + \{\psi_{j\longrightarrow m\longrightarrow i}\}$ denotes a set of transactions that directly and indirectly approve ψ_i, w_j is the weight of ψ_j, and any transaction counts once. Figure 6 demonstrates an example of how a transaction accumulates weight, where the rectangle represents transaction, and the number outside the parentheses in the rectangle represents transaction weight. The transaction c selects a and b which, respectively, are represented by the edges E_2 and E_3, so a and b both accumulate a direct approval, and their accumulative weight is added by the weight of c (it is 12 in Figure 6). b and c approve a directly; d and e approve a indirectly. If $w_b = 1$, $w_c = 12$, $w_d = 2$, and $w_e = 1$, then $\mathscr{A}_a = 5 + 1 + 12 + 2 + 1 = 21$.

4.4. Local Consistency Threshold. The existing DDLs own independent components to strictly sequence transactions, such as Tangle's coordinator [11]. Strictly ordered transactions are very important for financial applications to defend against the double attack. However, managing reputation does not need to be strict. We argue that a transaction, approved by enough but not be strictly ordered globally, is secure for auditing reputation for the reference nodes. DDL defines that a transaction reaches consensus when it achieves enough approval, making the transaction difficult to tamper. Combining the above discussion of the CW, it is easy to realize that the transaction accumulated a high CW which can be seen as secure and cannot easily be falsified. We need to find how much CW can ensure the security of transactions for auditing reputation.

The local consistency threshold is proposed to enable the ledger node to judge whether the transaction is secure by itself instead of relying on the third component. Before introducing the details of the local consistency threshold, let us clarify two essential and reasonable assumptions in this paper: (1) if two nodes do not meet within their trips (refer to the trip where they need the VANET applications), they do not need the reputation of each other. First, in terms of the collaboration, nodes only care about the nodes running in several nearby partitions; e.g., the traffic light data of a specific road section is only meaningful and can only be verified by the nearby nodes. Then, in terms of time, the node's behavior that is too old is no longer suitable for evaluating reputation for some security issue consideration [30]. (2) We consider that if a node has always been well-behaved in issuing a transaction (verification and selecting tips) and participating in the interaction (sharing and cooperating), then the ledger will eventually accept the transaction issued by this node with a high probability.

According to the above discussion, we set a period h, limiting all transactions' valid time. When a period ends, the ledger will be reset. We also define a set of transactions N_c, containing all the transactions for a node cared. However, the fewer nodes will lead to security risks for a distributed ledger, so we bring the workload of the nodes to increase the transaction's weight. Now, we can focus on the transaction consensus in $n(n \geq 1)$ partitions and define Θ as the local consistency threshold to assist nodes to infer the transaction's credibility, and it can be expressed by

$$\Theta = \frac{L}{|N_c|} \sum_{j\in N_c} \max_{k\in\{\psi_k\}^{(j)}} \left\{ w_k^{(j)} \right\}, \tag{6}$$

where $\{\psi_k\}^{(j)} = \{\psi_1, \psi_2, \cdots, \psi_k\}^{(j)}$ denotes the set of all the transactions issued by node p_j, $w_k^{(j)}$ is the weight of the $\psi_k^{(j)}$, so $\max_{k\in\{\psi_k\}^{(j)}} \{w_k^{(j)}\}$ presents the largest weight of the transaction issued by p_j. N_c represents a set of active nodes (issued transactions in a period) in n partitions cared about; $|N_c|$ is the number of these nodes. L is a positive hyperparameter that controls the evolution speed of the ledger, and the nodes can adjust it to cope with the scale change of the interest partitions. Nodes could make their judgments on whether the transaction is confirmed. Algorithm 1 introduces the detail of the transaction consensus process.

4.5. Tip Selection Algorithm. We propose a modification to the attachment mechanism of the Tangle. This modification ensures the transaction is verified and secure in the partitioning VANET and preserves essential features of the Monte Carlo Markov Chain (MCMC) selection algorithm [10].

Firstly, the issuing nodes need to verify whether or not the transaction selected for approval is mutually consistent with each other. If detecting an inconsistency, the tip selection process must be rerun until a consistent $\mathscr{L}(t)$ is found. In addition, creating m independent random walks in a path of DAG contains the transactions issued by the nodes running in the interested partitions in the current period. The walk starts at the genesis site and moves along the edges. The

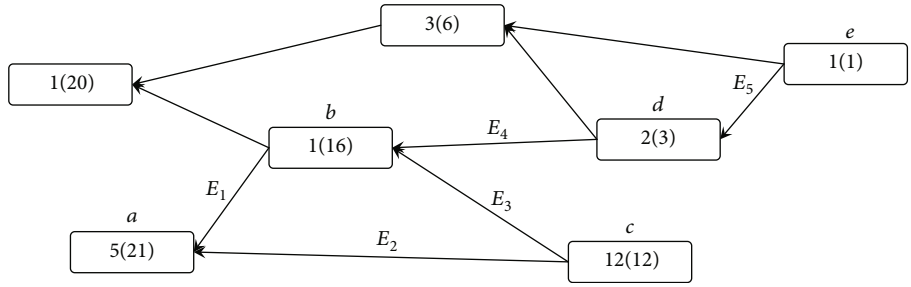


Figure 6: A part of ledger.

input: A tip ψ_i,local consistency thresholdΘ, current tip set $\mathscr{L}(t)$, sub-DAG consisted of the confirmed transactions G_t^c and sub-DAG consisted of the unconfirmed transactions G_t^u
output: Updated ledger $G_{t+1,}^c G_{t+1,}^u$ and the updated tip set $\mathscr{L}(t+1)$
1 Extract all approval transactions $\{\psi_{i\longrightarrow m}\}$packaged in $\{\psi_i\}$;
2 for each $\{\psi_{i\longrightarrow m}\}$**do**
3 calculate and add weight ω_i to ψi;
4 add ω_i to Cumulative Weight $\mathscr{A}_{i\longrightarrow m}$of $\psi i_{i\longrightarrow m}$;
5 if $\psi_{i\longrightarrow m} i\pi\ \mathscr{L}(t)$to G_t^u
6 add ψ_i to $\mathscr{L}(t)$to to G_t^u
7 move $\psi_{i\longrightarrow m}$from $\mathscr{L}(t)$to G_t^u
8 else
9 wait for a punish time;
10 add ψ_i to $\mathscr{L}(t)$;
11. Extract the transaction $\{\psi_{i\longrightarrow m\longrightarrow\eta}\}$that indirectly approved by
12 for each $\psi_{i\longrightarrow m\longrightarrow\eta}$ in $\{i_{i\longrightarrow m\longrightarrow\eta}\}$**do**
13 add w_i to $\mathscr{A}_{i\longrightarrow m\longrightarrow n}$ of $\psi_{i\longrightarrow m\longrightarrow n}$;
14 **if** $\psi_{i\longrightarrow m\longrightarrow n} > \Theta$ then
15 move $\psi_{i\longrightarrow m\longrightarrow n}$ to G_t^c
16 **final**;
17 **return** $G_{t+1}^c, G_{t+1}^c, \mathscr{L}(t+1)$;

Algorithm 1: Consensus algorithm for transactions on partitions.

Table 2: Data field in NSL.

Field	Symbol
Node ID	p_i
Iterative consensus metric	$\langle t, R^* \rangle$
Cumulative interactive metric	$\langle t, R^+, R^- \rangle$

probability of stepping along an edge from site ψ_j to site ψ_k is

$$f\left(-\alpha\left(\mathscr{A}_j - \mathscr{A}_k\right)\right), \tag{7}$$

where $f(\cdot)$ is an exponential function and α is a positive constant. $\mathscr{A}_j$ and $\mathscr{A}_k$ are the CWs of ψ_j and ψ_k, respectively. For a new transaction, suppose the walk should reach $Q(\mathrm{Q} \geq \mathrm{m})$ tips, and the issuing node selects the tip satisfying

$$\min_{l<Q}\left\{\sum_{i=1}^{m}\left|\Theta - \mathscr{A}_i^{(l)}\right|\right\}, \tag{8}$$

where $\mathscr{A}_i^{(l)}$ denotes the CW of transactions directly approved by ψ_l that is the end of a walk. Finally, we also need to walk to the m tips that their selected transactions are about to be or just recent security.

5. Reputation Update with Transaction Local Consistency

In this section, we first present the definition of each part of the vehicular reputation. Then, we will describe the reputation update method based on the partitioning and valid period.

5.1. Vehicular Reputation. Our DDL requires the vehicle to be a node and contributes to the ledger maintenance. Therefore, we argue that the expression of reputation needs to contain the node's behaviors in ledger maintenance and VANET interaction. Interactive behavior refers to the quality of data that the node shares when interacting, and it is stored in the relevant transactions issued after each interaction occurs. The maintenance behavior of the ledger, we also

input: *NSL*, a confirmed transaction ψ_i, issued node p_i and the node that waited to interact with p_j
output: The updated reputation μ of p_j
1/∗∗∗ Update NSL ∗∗∗/ Extract response nodes $\{p_{j\longrightarrow i}\}$ and corresponding interactive ratings $\{y_{j\longrightarrow i}\}$;
2 **for** each $\{p_{j\longrightarrow i}\}$ in $\{p_{j\longrightarrow i}\}$**do**
3 update $NSL(p_{j\longrightarrow i})[\langle t, R^+, R^- \rangle]$ based on the corresponding interactive rating $\{y_{j\longrightarrow i}\}$;
4 Extract R^* in ψ_i and update $\langle t, R_* \rangle$ to $NSL(p_i)$;
5/∗∗∗ Calculate Reputation ∗∗∗/ Obtain the current time t_u;
6 Calculate quality σ based on $NSL(p_j)[\langle t, R^+, R^- \rangle]$;
7 Select an exponential function to calculate $\eta = f(-\beta R^*)$;
8 Obtain the reputation μ;
9 **final**;
10 **return**μ;

Algorithm 2: Reputation update algorithm.

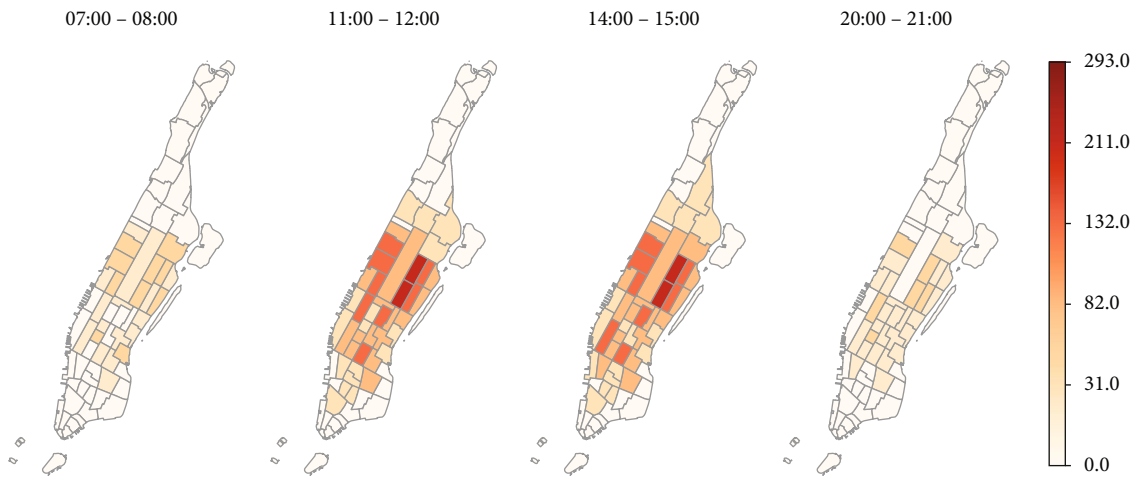


Figure 7: Distribution of the partitions and the vehicle heat in different hours in Manhattan.

call consensus contribution, refers to the node's performance in ledger maintenance and is calculated by other nodes when they verify a new transaction issued by the node.

In summary, when omitting the symbol of interest partitions and valid period, the reputation is

$$\mu = \tau_1 \sigma + \tau_2 \eta, \tag{9}$$

where σ and η represent the interactive quality and consensus behavior, respectively; we will discuss them in the following subsection. $\tau_1 + \tau_2 = 1$; they are used to adjust the ratio of the two measures in different scenarios. For example, in the initial phase of each valid period, the ledger needs as many as possible nodes to join to maintain the transaction consensus, at this case, $\tau_2 > \tau_1$.

5.1.1. Interactive Quality. To simply the discussion, we only consider the cooperating model because it can easily extend to the sharing model. Assume that a node has M_1-positive ratings and M_2-negative ratings, so that all the received ratings $M = M_1 + M_2$. Let $R_m^{(b+)}$ be the positive rating that p_b received at mth response at time t and $R_m^{(b-)}$ represent the negative rating. So we have

$$R_{t_u}^+ = \sum_{t=t_0}^{t_u} \sum_{m=1}^{M_1} R_m^{(b+)}, \tag{10}$$

$$R_{t_u}^- = \sum_{t=t_0}^{t_u} \sum_{m=1}^{M_2} R_m^{(b-)}, \tag{11}$$

where t_0 is the initial time of current valid period and t_u represents the current time. The interactive quality is calculated as follows:

$$\sigma_{t_u} = \frac{\theta_1 \cdot R_{t_u}^+ - \theta_2 \cdot R_{t_u}^-}{R_{t_u}^+ + R_{t_u}^-}, \tag{12}$$

where θ_1 and θ_2 are sensitivity weight; let $R^+ = R_{t_u}^+, R^- = R_{t_u}^-$, then

$$\theta_1 = \frac{F(R^+)}{F(R^+) + F(R^-)}, \theta_2 = \frac{F(R^-)}{F(R^+) + F(R^-)}. \tag{13}$$

$F(\cdot)$ represents the sensitivity function such as $F(x) = x$, $F(x) = x^2$, and $F(x) = x^3$; the sensitivity of the positive or

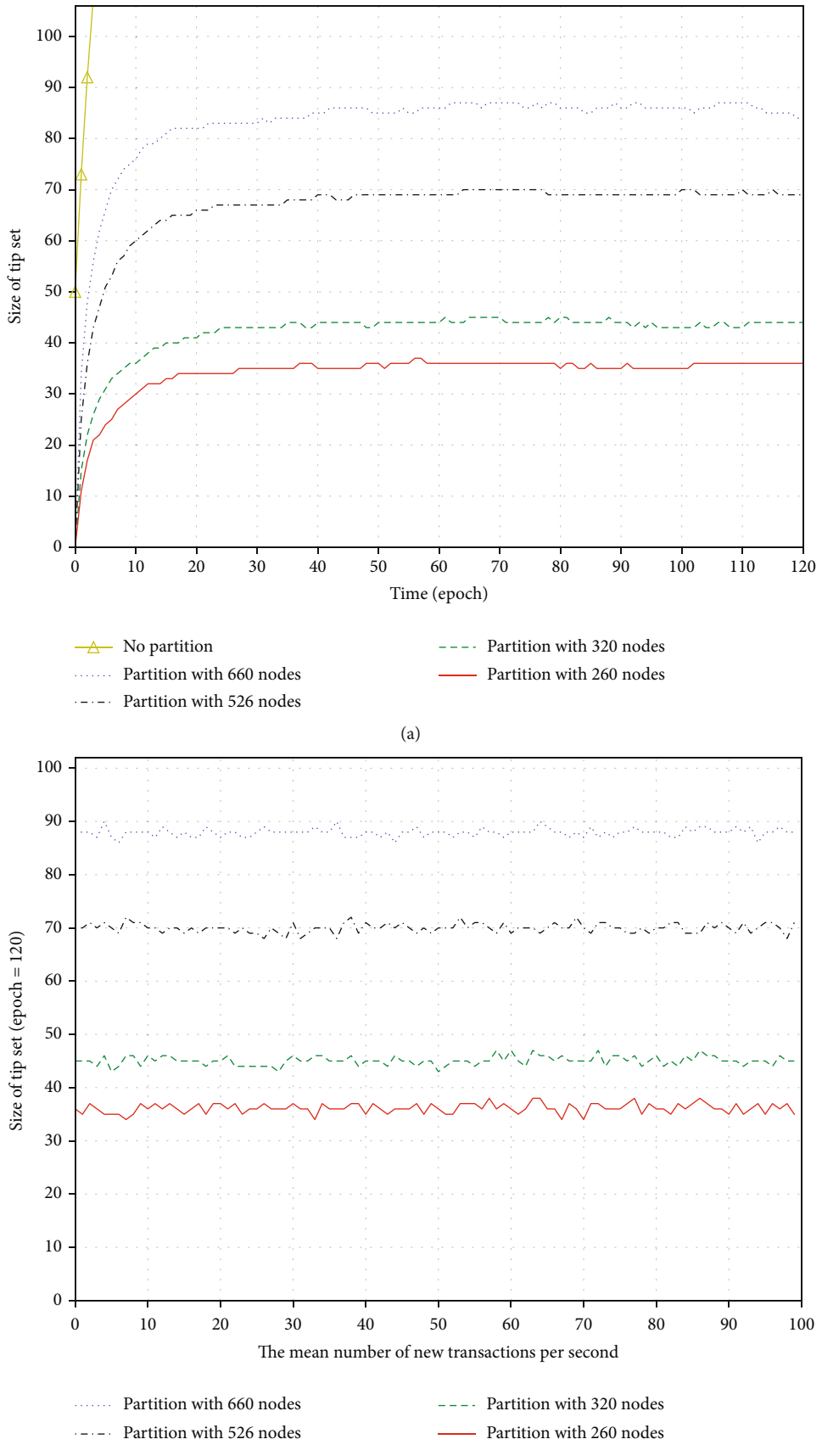


Figure 8: (a) The number of tips in partitions of different sizes. (b) The number of tips vs. TPS in partitions of different sizes, epoch = 120.

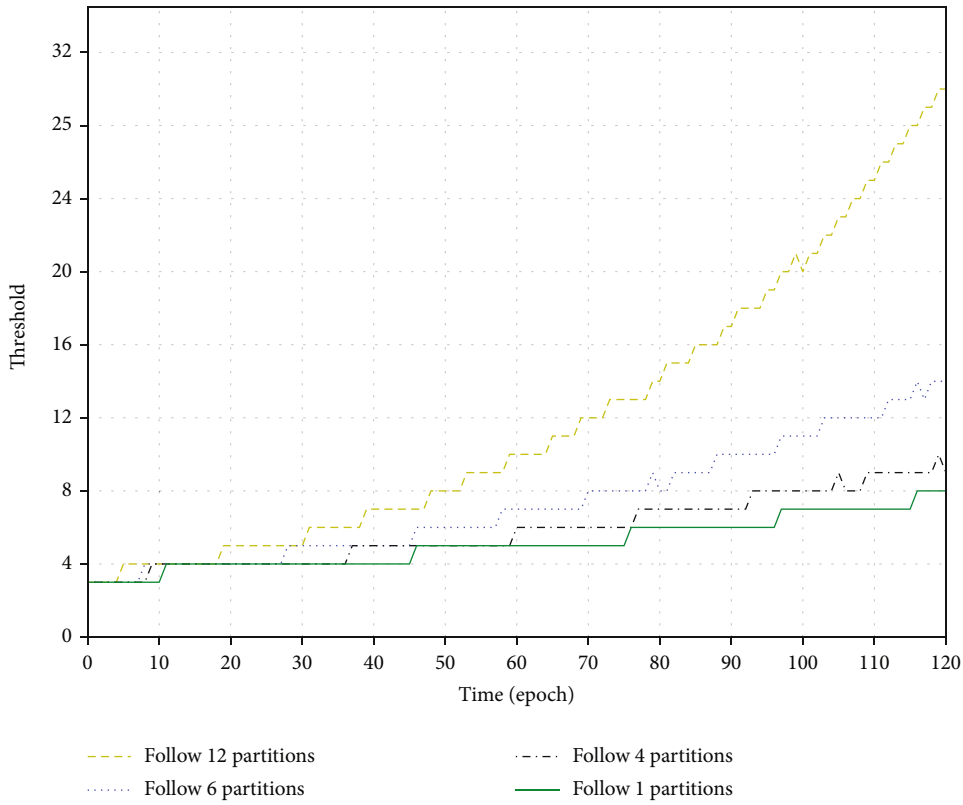


Figure 9: Local consistency threshold vs. epoch, $L = 1$.

negative rating in the metric of interactive quality can be controlled by θ. θ could adjust the weight of contribution rating based on the different requirements. For example, when the shared data may threaten the safety of humans, the weight of negative evaluation should be increased.

5.1.2. Consensus Contribution. When the node selects tips, its behavior determines its contribution to the consensus of DDL. The selections are represented by the edges and are public to all nodes, and they are important on ledger evolution. When receiving a new transaction ψ_i issued by p_b, a node needs to check the CWs of all the transaction in $\{\psi_{i\longrightarrow m}\}$ and calculate a consensus rating $R_{b*}(t)$ of p_b, and the calculation can be expressed by

$$R_b^*(t_u) = R_b^*(t_u - 1) + \sum_{m=1}^{m=M} (\Theta - \mathscr{A}_{i\longrightarrow m}(t_u)), \tag{14}$$

where $\mathscr{A}_{i\longrightarrow m}(t_u)$ presents the CW of the $\psi_{i\longrightarrow m}$ approved by ψ_i at time t_u. If $\Theta - \mathscr{A}_{i\longrightarrow m}(t_u) \le 0$, it denotes that the issuing node selected a confirmed transaction, which represents a bad behavior, of course; otherwise, it denotes a tip is selected or there is other issued unconfirmed transaction recently, which means a good behavior. Therefore, if the node performs positively, then $0 \le R_b^*(t_u) \le 1$; otherwise, $-1 \le R_b^*(t_u) \le 0$. The iterative method is used because the CW of a transaction always increases along with time. Define the metric of consensus contribution as

$$\eta = f(-\beta R_b^*), \tag{15}$$

where β is an attenuation factor; it can be adjusted with the ledger network change even if node p_b does nothing in a time interval. Thus, the metric of consensus contribution of a node's reputation can be calculated by auditing the local transactions at any given time.

5.2. Reputation Update. Since the dimension of the DDL grows with time, it would be nonfeasible to search and audit the related transactions for reputation calculation even in one partition. A possible solution is to maintain additional data structures to store intermediate reputation calculations to save the computing power and time required for transaction search. We called this additional data structure as Node Status List (NSL). Each node should initialize an NSL when first connecting the ledger or a new valid period starting. Table 2 presents the data field contained in each row of the list. The first column is node ID. The second is a set of tuples, and the elements recorded the calculating timestamp and consensus metric. The third is a tuple containing the interactive metric and the update timestamp.

Now, we consider the reputation update method based on the intermediate data structure. The node can calculate the transaction weight and the vehicular reputation by searching the NSL. If a new transaction passes the verification or exceeds the threshold, each receiving node will update the specific field at local. Algorithm 2 describes the transaction consensus with node status data update.

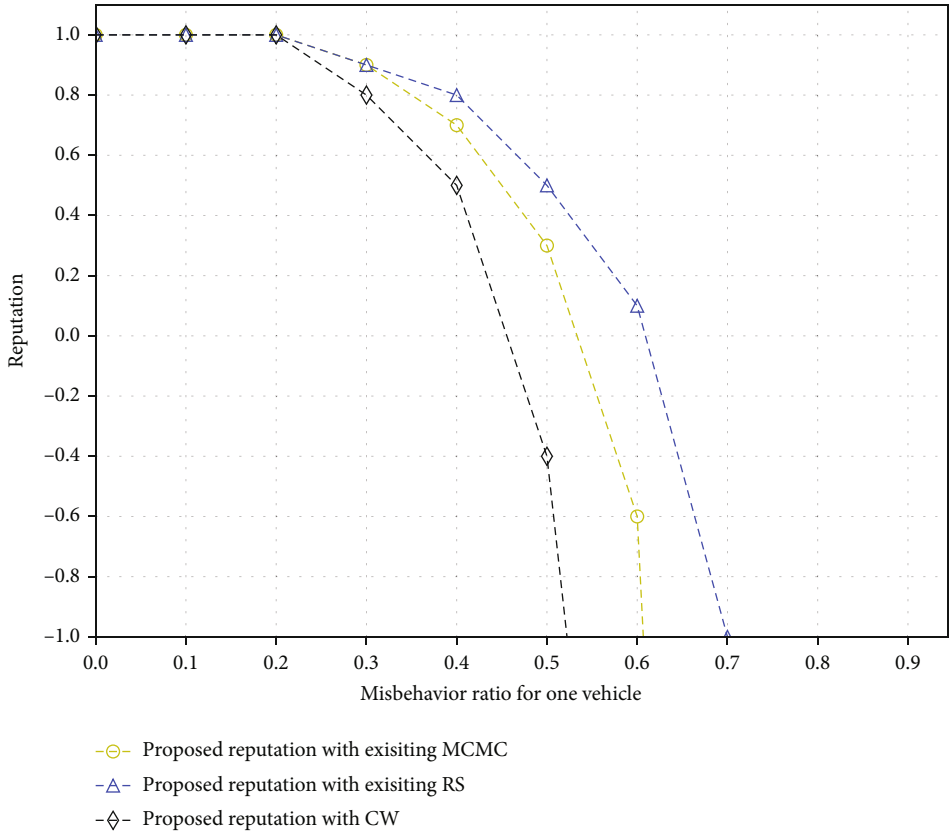


Figure 10: Reputation vs. misbehavior ratio for one vehicle, $f(\cdot) = x^3$.

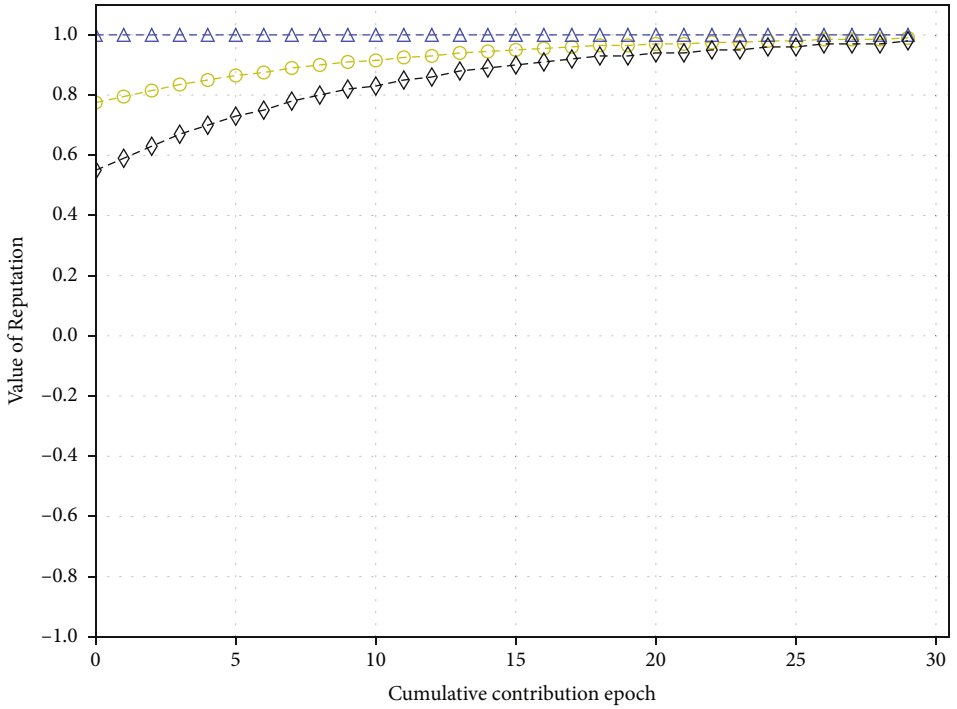
6. Experimental Result and Analysis

We build a trip set based on the New York yellow cab [31] and extract the trajectories at daytime of several days in December 2020. The vehicle number in the evaluation scenario is a uniform distribution between 5800 and 7900 and distributed in 55 partitions. All the partitions are fixed according to predivided partitions in [31], and we filter partitions with less than 20 vehicles and merge these vehicles into nearby partitions. Figure 7 shows the distribution of the partitions and vehicles' heat at different times of one day. We can find that vehicle numbers in different partitions at the same period are very different, and so do the number of vehicles in the same partition at different periods.

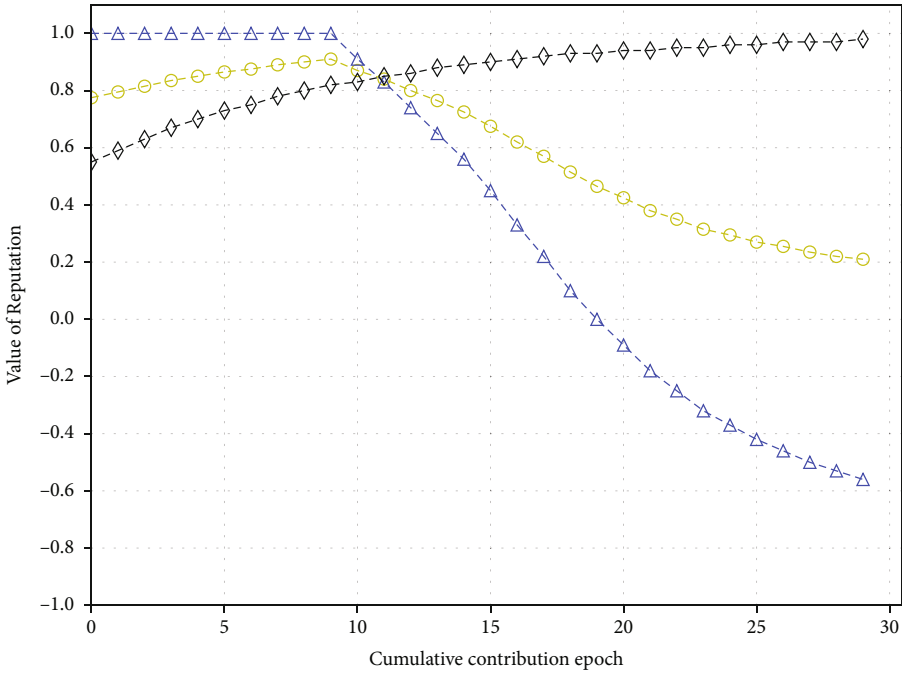
In addition, the entry of vehicles into VANET follows the Poisson process with an average of λ [32], so does the arrival of transactions. Assume that vehicles issuing the transactions follow the power-law distribution [33, 34] and the number of the reference vehicles for the interaction is a uniform distribution between 3 and 10. Considering the simulation is conducted in daytime, we set the system's throughput per second (TPS) in a partition to be large and positively correlated with the number of nodes. Suppose the global arrival rate $\lambda_{all} \leq 1000$, and for each partition, its arrival rate λ_k is also allocated according to the proportion of vehicles owned by it. Thus, for any partitions in our evaluations, suppose $3 < \lambda_k < 86$.

The simulation is implemented using Python 3.8.3 in Windows 10 system with a Lenovo laptop, which has four cores and 32 GB memory. Cryptography is the Python cryptography library (v2.8) [35] and the Hashlib standard library (3.7.7) [36].

6.1. Convergence of the Proposed Ledger. We first investigated the convergence performance of our proposed DAG-based distributed ledger. Figure 8(a) shows that the size of $\mathscr{L}(t)$ increases quickly in the first few epochs and reaches a stable state after around 20 epochs. The main reason for the rapid accumulation of tips in the early stage is the waiting period u. After a tip is released, it will take a while to be "seen" and verified by the nodes. Meanwhile, when a new validity period begins, the node's reputation and each metric are reset to 0.5, so the tips' weights and the parent's selection are very close. The transaction selection can be thought of as random in all transactions, which will cause some tips to be unable to be verified in time. However, as the number of epochs increases, the size of $\mathscr{L}(t)$ becomes stable, which verifies the convergence performance of our proposed DAG ledger in the case of the partitioning method. In Figure 8(b), we observe the change of the size of $\mathscr{L}(t)$ around 120th epoch though adjusting the TPS in partitions. It can be seen that the convergence performance will not be affected because of the definition of the tip and the verifying-before-issuing mechanism of the transaction; that is, when a new transaction is

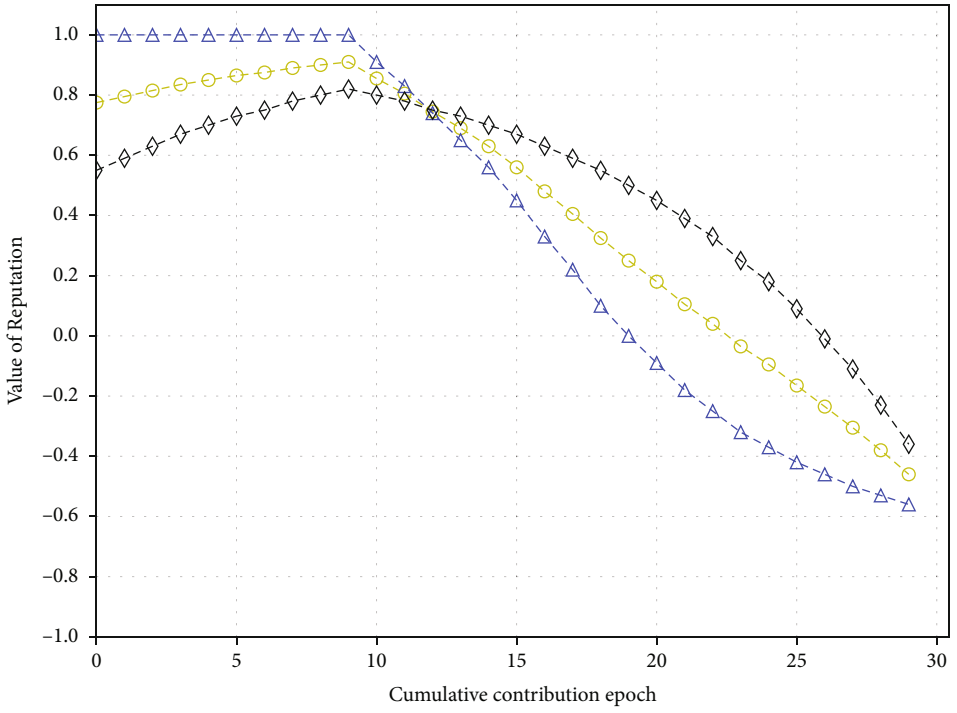


(a) Honest node. The node reputation is reset to 0.5 at the beginning of each validity period. When the node starts to contribute, the evaluated reputation accumulate from 0

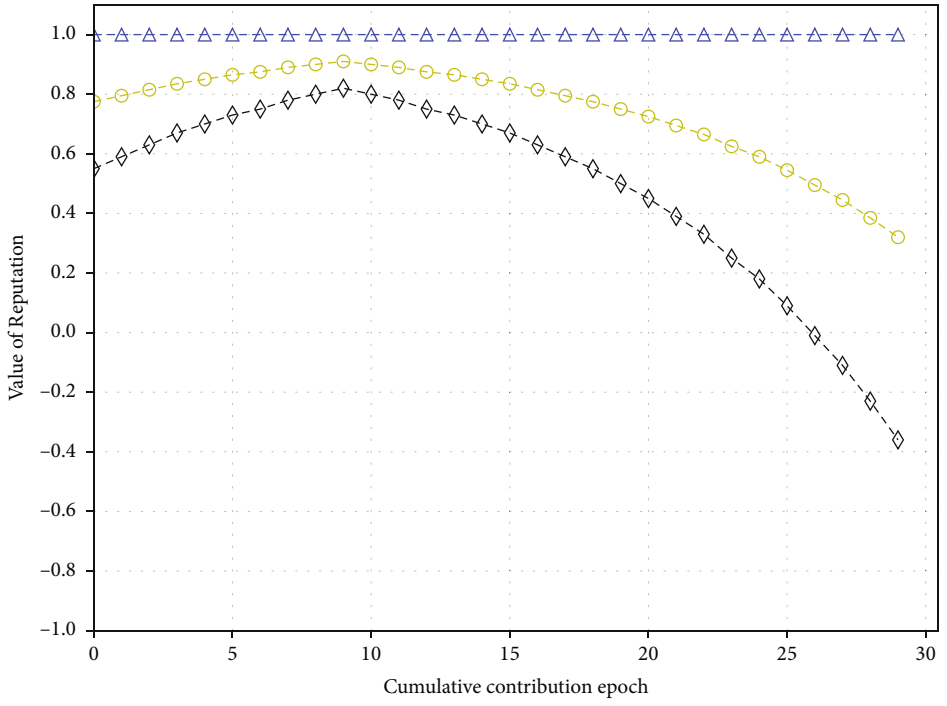


(b) Corrupted node (from epoch = 10), the bad behavior only involves sharing part; the consensus part is normal

Figure 11: Continued.



(c) Corrupted node (from epoch = 10), the bad behavior only involves consensus part; the sharing part is normal



(d) Corrupted node (from epoch = 10), the bad behavior involves both sharing and consensus parts

Figure 11: Reputation accumulation process, $\tau_1 = \tau_2 = 0.5$, $f(\cdot) = x^3$.

added, several tips need to be verified. Therefore, the increase of TPS will also improve tip verification. So, the convergence performance of our proposed DAG ledger in the case of the partitioning method was verified.

6.2. Local Consistency in Multipartitions. Next, we investigated the performance of the local consistency threshold. Figure 9 shows the results of the threshold changing with the number of the followed partitions. We also set the node's participation following the power-law distribution, and only the top 20% active nodes are considered when calculating the threshold starting from the middle of a period (randomly from 40 to 60 epochs in each evaluation). The results show that the threshold increase is slow in the initial few epochs. This is because the threshold is positively correlated with the reputation of the active nodes, and it will be reset to 0.5 in the initial stage of a new validity period. Furthermore, reputation always rises slowly at the beginning of the validity period. Even the most active nodes also need to spend multiple epochs to conduct one good behavior (answering the request from others or computing the PoW for approving transactions) and reach consistency with other related nodes.

Note that in the later epochs, the threshold grows slowly at the scenario of followed fewer partitions, and this is because many nodes have left the followed partitions before their reputation accumulated high enough. However, the reputation grows faster in the scenario of following more partitions, and this is because we can observe the nodes in a more extensive range (involved more partitions), so the nodes have enough time to accumulate a sufficiently high reputation. Moreover, we summarize that some nodes (a taxi will operate for a long time and drive within some fixed partitions) can accumulate much and soon based on the power-law distribution. Therefore, the local consistency threshold is effective; short-travel nodes only need to pay attention to fewer partitions and the recent behavior of the nodes, while long-travel nodes need to pay attention to more partitions and the long-term behavior of the nodes.

6.3. Performance of Reputation Update. Then, we test the resiliency of our reputation representation against misbehavior of vehicles in Figure 10. The misbehavior includes bad collaboration performance (obtained a lower shared rating) or selecting an old and approved transaction to attach. In Figure 10, we can see that all the schemes with different tip selection algorithms can reduce the node's reputation below 0.5 when the node's misbehavior ratio exceeds 50%. Besides, when the reputation is lower than 0.5, the decline is very fast, mainly caused by the bad consensus behavior. Typically, if a node becomes lazy, it would select a fixed transaction set to save its computational power. Meanwhile, the CW of any approved transaction inevitably increases (raised by the indirect approval) with epoch, and the exponential function in Formula (5) also speeds up the decline. This also provides incentives to the node to select the tip.

Last, Figure 11 shows the impact of different metrics on reputation when bad behavior accumulates. We can conclude that when there is bad sharing behavior, the reputation begins to decline rapidly, while the consensus become bad, and the downward trend is slow. This is mainly controlled by the setting of hyperparameter. We can strengthen the weight of consensus metric by adjusting the proportion of τ For example, if the sensitivity function $f(\cdot) = x$, then the sharing metric will decrease linearly. $f(\cdot) = x^3$ can reduce some misbehavior caused by inevitable communication delay; sharing metric will begin to decline rapidly after the misbehavior exceeds the tolerance limit. In addition, when there are few numbers of the activated vehicles, the reputation system can increase the weight of the consensus metric by recommending a large τ_2, so as to attract more vehicles to help verify new transactions, which is also to accumulate its own reputation (consensus metric).

7. Conclusion

This paper proposes a partitioned DDL for maintaining the vehicular reputation to support the trust establishment in VANET. We design the transaction for the vehicular reputation auditing using the details of interactions among vehicles. To encourage the vehicle to maintain the ledger, we design a vehicular reputation evaluation method by aggregating the contribution in vehicular interaction and ledger consensus maintenance. Besides, a reputation update method based on the consistency of transactions in one or several partitions is presented to allow any vehicle to evaluate other's reputations anywhere and anytime. Simulation results demonstrate that our partitioned DDL is practical in real-world scenarios and achieves a better detection rate of bad behavior than the baselines with various tip selection algorithms. Future work is in progress to consider how to partition the VANET better to improve the vehicle's safety during its trip.

Data Availability

The vehicle tip data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Lu, J. Zhang, Y. Qi et al., "Accelerating at the edge: a storage-elastic blockchain for latency-sensitive vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021.

[2] S. Smys and H. Wang, "Security enhancement in smart vehicle using blockchain-based architectural framework," *Journal of Artificial Intelligence*, vol. 3, no. 2, pp. 90–100, 2021.

[3] C. Pu, "A novel blockchain-based trust management scheme for vehicular networks," in *2021 wireless telecommunications symposium (WTS)*, pp. 1–6, CA, USA, 2021.

[4] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in internet of vehicles with blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11815–11829, 2020.

[5] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium

Blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[6] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[7] S. Nakamoto and A. Bitcoin, *A peer-to-peer electronic cash system*, 2008, https://bitcoin org./bitcoin.pdf.

[8] S. King and S. Nadal, *PPCoin: peer-to-peer crypto-currency with proof-of-stake*, self-published paper, 2012.

[9] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[10] S. Popov, *The tangle*, White paper, 2018.

[11] Coordicide Team, IOTA Foundation, *The coordicide*, 2019, https://files.iota.org/papers/Coordicide WP.pdf.

[12] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[13] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi, "A framework for secure vehicular network using advanced blockchain," in *2020 international wireless communications and Mobile computing (IWCMC)*, pp. 1260–1265, Limassol, Cyprus, 2020.

[14] P. C. Bartolomeu, E. Vieira, and J. Ferreira, "IOTA feasibility and perspectives for enabling vehicular applications," in *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.

[15] Y. Lu, Y. Qi, and S. Qi, "Say no to price discrimination: decentralized and automated incentives for price auditing in ride-hailing services," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[16] Y. Lu, J. Zhang, Y. Qi et al., "Safety warning! Decentralised and automated incentives for disqualified drivers auditing in ride-hailing services," *IEEE Transactions on Mobile Computing*, vol. 21, p. 1, 2021.

[17] Z. Lu, Q. Wang, G. Qu et al., "BARS: a blockchain-based anonymous reputation system for trust management in VANETs," in *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, pp. 98–103, New York, NY, USA, 2018.

[18] N. Szabo, *Smart contracts*, 2014, http://szabo.best.vwh.net/smart.contracts.html.

[19] H. Liu, D. Han, and D. Li, "Fabric-IoT: a blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.

[20] D. Wang, L. Zhang, C. Huang, and X. Shen, "A privacy-preserving trust management system based on blockchain for vehicular networks," in *2021 IEEE wireless communications and networking conference (WCNC)*, pp. 1–6, Nanjing, China, 2021.

[21] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[22] S. Kudva, S. Badsha, S. Sengupta, H. la, I. Khalil, and M. Atiquzzaman, "A scalable blockchain based trust management in VANET routing protocol," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 144–156, 2021.

[23] A. Khalid, M. S. Iftikhar, A. Almogren, R. Khalid, M. K. Afzal, and N. Javaid, "A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs," *Information Processing & Management*, vol. 58, no. 2, p. 102464, 2021.

[24] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[25] F. Li, Z. Guo, C. Zhang, W. Li, and Y. Wang, "ATM: an active-detection trust mechanism for VANETs based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4011–4021, 2021.

[26] E. Diallo, O. Dib, and A. K. Al, "An improved PBFT-based consensus for securing traffic messages in VANETs," in *2021 12th international conference on information and communication systems (ICICS)*, pp. 126–133, Valencia, Spain, 2021.

[27] X. Zhang, R. Li, W. Hou, and H. Zhao, "V-Lattice: a lightweight blockchain architecture based on DAG-lattice structure for vehicular ad hoc networks," *Security and Communication Networks*, vol. 2021, 17 pages, 2021.

[28] C. LeMahieu, *Nano: a feeless distributed cryptocurrency network*, 2018, https://nano. org/en/whitepaper.

[29] A. Churyumov, *Byteball: a decentralized system for storage and transfer of value*, white paper ed edition, , 2016https://byteball.org/Byteball.pdf.

[30] Q. Li, A. Malip, and K. M. Martin, "A reputation-based announcement scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, 2012.

[31] *New York City Taxi and Limousine Commission*https://www1.nyc.gov/site/tlc/about/data.page.

[32] Q. Cui, N. Wang, and M. Haenggi, "Vehicle distributions in large and small cities: spatial models and applications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10176–10189, 2018.

[33] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Exponential and power law distribution of contact duration in urban vehicular ad hoc networks," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 110–113, 2012.

[34] T. Chen, Z. Li, Y. Zhu et al., "Understanding ethereum via graph analysis," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 2, pp. 1–32, 2020.

[35] *Cryptography library (v2.8)*https://cryptography.io/en/latest/.

[36] *Hashlib standard library (3.7.7)*https://docs.python.org/3/library/hashlib.html.