# A Decentralized Trust Management System for Intelligent Transportation Environments

Xiao Chen, Jie Ding, and Zhenyu Lu

*Abstract*—**Commercialized 5G technology will provide reliable and efficient connectivity of motor vehicles that could support the dissemination of information under an intelligent transportation system. However, such service still suffers from risks or threats due to malicious content producers. The traditional public key infrastructure (PKI) cannot restrain such untrusted but legitimate publishers. Therefore, a trust-based service management mechanism is required to secure information dissemination. The issue of how to achieve a trust management model becomes a key problem in the situation. This paper proposes a novel prototype of the decentralized trust management system (DTMS) based on blockchain technologies. Compared with the conventional and centralized trust management system, DTMS adopts a decentralized consensus-based trust evaluation model and a blockchain-based trust storage system, which provide a transparent evaluation procedure and irreversible storage of trust credits. Moreover, the proposed trust model improves blockchain efficiency by only allowing trusted nodes participating in the validation and consensus process. Additionally, the designed system creatively applies a trusted execution environment (TEE) to secure the trust evaluation process together with an incentive model that is used to stimulate more participation and penalize malicious behaviours. Finally, to evaluate our new design prototype, both numerical analysis and practical experiments are implemented for performance evaluation.**

*Index Terms*—**Trust management, vehicular network, blockchain, TEE, consensus protocol, incentive mechanism.**

## I. Introduction

INTELLIGENT transportation system will gain benefits from the commercial use of 5G communication technology in the near future. Then, most new vehicles can support diverse information services, including safety, entertainment and even social applications. As a huge amount of information can be independently shared among vehicular networks, the vehicles behave as human social activities in terms of their application services. Thus, these interconnections can be considered a "social network of intelligent vehicles" since each driver can share data with other neighbours. Thus, the concept of "vehicular social networks (VSNs)" is coined [1]. A VSN could be defined in a broader sense as a network of physically or virtually connected vehicles that are interested in sharing information for a common purpose or benefit [2].

Nevertheless, with the involvement of human factors in vehicular networks, social characteristics and human behaviours largely impact information dissemination over the network. Thus, it is significant for drivers to distinguish trustworthy from untrustworthy information. Most previous research relies on traditional security technologies (e.g., PKI) to build the defence for illegitimate vehicles [3]. However, it is still possible for legitimate vehicles to send untrustworthy information due to selfish or malicious intents. Hence, trust-based solutions are considered to eliminate dishonest vehicular nodes [4].

Due to the decentralized architecture of VSNs, establishing trust management is still a challenge. Trust evaluation and management are usually based on the nodes' behaviours, such as interactions and participation in a community. Thus, the trust management system enables the evaluation of vehicular nodes by rating the information they published, and confines the false information being shared over the network [5]. Most trust management systems are designed in a centralized architecture, e.g., Refs. [6]–[8]. Such centralized trust management systems have all rating operations and results implemented and preserved in central servers, such as a trusted third party, which is not transparent to users. It still has a potential risk of leveraging a centralized third party after the accident of the Edward Snowden leaks [9], [10]. It is also challenged by a growing amount of sharing information and evaluation demand relying on limited central resources. Hence, it becomes a key concern to develop a new trust management prototype for higher security and performance demands.

Considering this information altogether, a decentralized trust management system (DTMS) is conceived based on the trust evaluation model, incentive model and blockchain-based consensus model. First, the trust evaluation model is designed along with a decentralized evaluation structure in which trust credits are calculated with a group of local nodes rather than

Manuscript received September 9, 2019; revised March 15, 2020 and June 3, 2020; accepted July 15, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61702233, Grant 61472343, and Grant 61773220; in part by the State Key Laboratory of Software Development Environment under Grant SKLSDE-2017KF-03; and in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu under Grant 18KJB520052. The Associate Editor for this article was Q. Zhang. *(Corresponding authors: Jie Ding; Zhenyu Lu.)*

Xiao Chen was with the Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science and Technology, Nanjing 210044, China. He is with the School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB, U.K. (e-mail: xiao.chen@ed.ac.uk).

Jie Ding is with the China Institute of FTZ Supply Chain, Shanghai Maritime University, Shanghai 200135, China, and also with the School of Mathematical Science, Yangzhou University, Yangzhou 225002, China (e-mail: jding@shmtu.edu.cn).

Zhenyu Lu is with the School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: luzhenyu76@163.com).

Digital Object Identifier 10.1109/TITS.2020.3013279

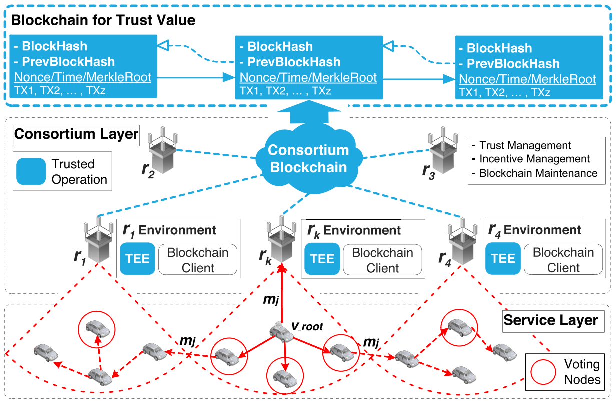


Fig. 1. Scenario of decentralized Trust management framework.

a third party. Second, the incentive model will stimulate more node participation in rating and validating activities by gaining extra rewards (e.g., trust credits), and penalize the nodes with malicious behaviours. Finally, the obtained trust credits and incentive rewards are recorded in the form of transactions that must be validated and agreed upon via the consensus step before uploading to the blockchain. The blockchain mainly contributes to its transparent data management without relying on a third party, and ensures traceability and irreversibility of records, in contrast to those for third parties. In addition, the blockchain-based consensus mechanism provides a solution to fix the Byzantine Generals Problem [11], which means to tolerate any adversarial nodes during the consensus process and to guarantee an eventual agreement and update on the confirmed trust credits.

Moreover, DTMS is designed to meet both security and performance challenges through its hardware-secured (i.e., trusted execution environment, TEE) model design and a parallel system architecture design. TEE is assumed to aggregate in base stations, as shown in Fig. 1. TEE achieves the isolation of data and operations in a secure enclave that allows its access only through an attested link supported by hardware-secured technologies (e.g., Intel software guard extensions, Intel SGX). Conversely, to improve performance without compromising security, DTMS adopts a parallel design of models by setting the system in a hierarchical structure with two layers. In Fig. 1, the bottom service layer supports operations of message rating and blocks validating; the top consortium layer maintains the evaluation of trust and incentive, as well as the consensus on the block validity. Such a hierarchical design achieves the parallelism of model operations and enhances the scalability and performance of the system. According to the key features of DTMS, the main contributions can be highlighted as follows:

1) A novel DTMS is proposed for trust management under a distributed intelligent transportation environment, which leverages TEE and blockchain technologies to build a secure and transparent trust evaluation and storage mechanism.
2) The trust model in DTMS can generate an accurate trust evaluation by using appropriate evaluators based on their properties and trust credits.
3) The incentive model is designed to assist the quality of services by providing necessary rewards or penalty to participants.
4) DTMS employs TEE and multi-signature technologies to secure trust evaluation and blockchain consensus, respectively, for higher security but lower communication overhead.
5) DTMS is designed in a hierarchical structure for better scalability and performance, especially the novel design of a parallel consensus model that provides superior performance due to its high scalability.

The rest of paper illustrates the related research, detailed designs and models, and performance evaluation analysis.

## II. Related Work

The issue of trust management in a vehicular scenario has been discussed by many researchers in past years. Yang and Wang [2] proposed a social network approach to study trustworthy information sharing in a vehicular network, which discussed how to apply traditional trust models used in online social networks to vehicular social networks. Moreover, the trust issue is also an important concern in the area of IoT. For instance, Yan *et al.* [4] investigated the properties of trust, proposed objectives of IoT trust management, and provided a survey on current literature advances towards trustworthy IoT. Due to the significance of the trust issue in a wide range of application domains, most previous research has focused on centralized trust management of vehicular networks or IoT systems.

### A. Centralized Trust Management

For the trust in vehicular networks, Li and Song [7] proposed an attack-resistant trust management scheme (ART) that was able to detect and cope with malicious attacks and evaluate the trustworthiness of both data and mobile nodes in VANETs. In this scheme, data trust was evaluated based on the data sensed and collected from multiple vehicles; thereafter, the node trust degree was calculated in two dimensions: functional trust and recommendation trust. Kerrache *et al.* [8] proposed a novel trust establishment architecture, called T-VNets, that was fully compliant with the ETSI ITS standard and took advantage of the periodically exchanged beacons (i.e., CAM) and event-triggered messages (i.e., DENM). This solution allowed the estimation of the traffic density, trust among entities, as well as the dishonest node distribution within the network. Additionally, researchers also considered as a trust issue the construction of reputation systems. For example, Li *et al.* [13] developed a novel announcement scheme for vehicular networks, in which a reputation system was applied for the evaluation of message reliability. Lai *et al.* [14] and Zhang [15] researched a reputation-based incentive scheme, respectively, which aimed to encourage cooperation and punish malicious vehicles and benefited the performance of the non-cooperative equilibria emerging in such applications.

### B. Decentralized Trust Management

Most previous research, such as all the abovementioned literature, contributed to the centralized architecture in which there were inherent advantages such as a single control point, availability, etc. However, according to the current VSN scenario, we cannot imagine a trust management system based only on a centralized architecture. For a distributed scenario, the decentralized system architecture is introduced for trust management. Huang *et al.* [16] presented a decentralized situation-aware trust architecture for the vehicular network. In Ref. [17], the authors proposed a data-centric trust management scheme for ad hoc networks, in which each node first calculated the trust credits in a distributed way and then aggregated the credits through a specific algorithm. Similarly, in Moustafa's work [18], a stigmergic-based approach was proposed to model the decentralized service interactions and handle the service composition in highly dynamic open environments, as the traditional centralized service composition techniques could not meet the needs of applications in decentralized environments, such as VSNs or IoT systems. Moreover, Li studied joint privacy and reputation assurance for vehicular ad hoc networks. This scheme adopted a decentralized reputation management model, in which the reputation assessment of each node was collectively performed by itself and its neighbours [19]. Additionally, decentralized trust management was utilized for other areas, such as cloud services. CloudArmor was a reputation-based trust management framework that provided a set of functionalities to deliver trust as a service (TaaS), which also managed the availability of the decentralized implementations of the trust management service [20].

The literature introduced thus far are all related to a decentralized architecture to support trust evaluation, which fulfils the requirements of the distributed application scenario. However, all these decentralized schemes are not complete-decentralized, as most schemes require the assistance of central servers to complete the final calculation or maintenance of trust credits. Thus, a trusted third party is still required in such trust management systems. However, in the situation in which third parties are not always trusted by all users, a fully complete-decentralized trust management architecture must be incorporated. With this in mind, some researchers have initiated a new direction of decentralized trust management by introducing blockchain technologies.

### C. Blockchain-Based Decentralized Trust Management

Yang *et al.* [21] proposed a decentralized trust management system in vehicular networks based on blockchain techniques. In Yang's solution, roadside units (RSUs) calculated the trust credit offsets of the involved vehicles based on evaluation results from vehicles and packed these data into a block. Thereafter, each RSU competed to add its holding blocks to a blockchain that was maintained by employing the joint proof-of-work and proof-of-stake consensus mechanism. In this scheme, all RSUs collaboratively maintained an updated, reliable, and consistent trust blockchain. However, this scheme overlooked an important problem. Generally, the consensus mechanism is collaboratively implemented by participants with conflicting interests; nevertheless, in a city or local areas, RSUs are usually deployed and maintained by one or two network service providers, which causes potential security risks (e.g., 51% attack). Furthermore, some other researchers have worked on new consensus protocols that could improve the efficiency of blockchain maintenance. Zou *et al.* [22] presented a novel consensus protocol called the proof-of-trust (PoT) consensus that was suitable for the crowdsourcing as well as the general online service industry. Although the PoT protocol avoided the low-throughput and resource-intensive pitfalls associated with Bitcoin's proof-of-work (PoW) mining, it is still risky when completely discarding the PoW. In addition to consensus mechanism research, researchers have investigated how to introduce blockchain to IoT applications. Li [23] explored the transactions in IoT systems and how they were processed; Roos [24] and Lind [25] considered efficient decentralized off-chain transfer but using different approaches that (the software-based payment channel and hardware-based trusted execution environment). He *et al.* [26] studied a blockchain-based truthful incentive mechanism for distributed P2P applications, and Sharma *et al.* [27] provided a secure distributed fog node architecture based on software-defined network (SDN) and blockchain techniques.

According to this literature, blockchain can be widely applied to various applications, in which a key issue is how to achieve fast operations on the blockchain. In our previous research, a decentralized trust management scheme was developed for VSNs [28]; however, this scheme still stored the data in central cloud servers. To address this issue, this paper explores a blockchain-based decentralized trust management framework. The details of the proposed scheme will be illustrated in the following sections.

## III. Blcokchain-Based Decentralized Trust Management Framework Scenario

### A. Blockchain

A *blockchain* is a growing list of records, named *blocks*, which consist of a cryptographic hash of the previous block, time stamps, and transaction data; these blocks are linked with cryptography. The first blockchain was conceptualized and utilized for Bitcoin by Satoshi Nakamoto in 2008. The blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across a peer-to-peer network, in which the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network. The blockchain is managed autonomously by the peer-to-peer network and a distributed timestamping server, which are authenticated by mass collaboration powered by collective self-interests. The blockchain achieves consensus on the transactions without a central instance and mutual trust relationships between participants [29].

Generally, the blockchain works towards a consensus in a distributed manner, which tackles the Byzantine Generals Problem from a practical angle. Since the blockchain has provided a feasible way to keep data security and consistency without a trusted central party, it can be utilized for a decentralized trust management scenario.

### *B. Decentralized Trust Management System*

In information system and information technology, *trust management*, introduced by Matt Blaze *et al.* [30], is an abstract system that processes symbolic representations of social trust, usually to aid the automated decision-making process. Previously, a single globally trusted server determines the trust credits of every node in the network, which is known as *centralized trust management*. However, global trust management restrains the reliability and accuracy of trust credits by evaluating them based on the decision of central servers. To address this issue, this paper will present a new decentralized trust management system by replacing the globally trusted server with a consensus of the most participating nodes based on a blockchain, in which each "transaction" represents either a node trust credit or a node incentive value.

Fig. 1 represents the architecture of the proposed DTMS, which consists of two separated layers: the *consortium layer*, includes a set of base station nodes that execute most operations of defined models and algorithms and maintain the blockchain; and the *service layer* includes a large number of vehicle nodes that act as either customer (i.e., message senders) or service providers (i.e., trust evaluators or blockchain validators). According to the design of DTMS, in the first trust evaluation stage, base stations play the main role in the consortium layer, which undertake the organization of trust evaluation and maintenance of the blockchain. Once a new trust-evaluation round starts, the base station first locks all vehicles under its coverage at this moment and then collects all messages during the past time segment. DTMS can group all messages by their properties (e.g., traffic information, entertainment advertisement, etc.) for the following evaluation steps. All sorted messages are shared with other base stations to recruit raters and continue evaluation operations at each base station. Then, each base station collects all the rating results from other stations and itself to calculate the eventual global trust credit of each vehicle node-locked by the station. The second incentive stage implements either rewards or punishments to both message senders and evaluators, in terms of their behaviours in the evaluation round. In the last consensus stage, once all global trust credits are generated, the base stations select a group of trusted nodes from a candidate set as the validators of this round. Once the validation and agreement are achieved among most validators, the global trust credits will be updated on the blockchain.

In the following section, models included in DTMS will be demonstrated based on the VSN scenario.

## IV. Design of DTMS

In Fig. 1, the decentralized trust management framework is implemented on a blockchain that is managed by a consortium network and supported by a set of selected mobile nodes based on a designed criterion. According to the design scheme represented in Fig. 2, the DTMS contains three key models: trust evaluation model (TEM), consensus model (CM), and incentive model (IM). It also includes two types of nodes that are vehicle nodes (i.e., $v_i$s) and base station nodes (i.e., $r_k$s). The vehicle nodes compose the service layer shown in Fig. 1 and can behave as message senders, trust evaluators or blockchain validators depending on their participant activities in the system. The base station nodes, running in the consortium layer, undertake the most execution of models in a TEE environment collaborating with vehicle nodes.

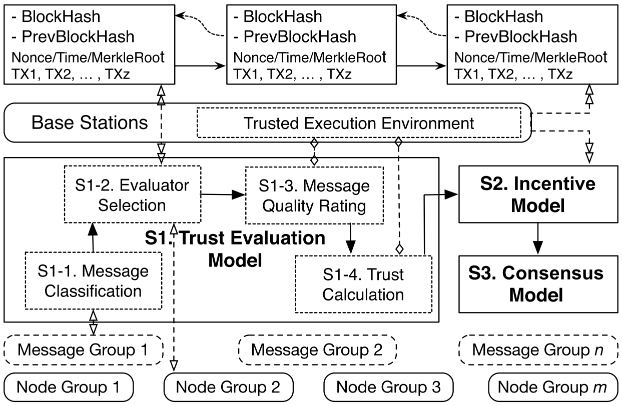


Fig. 2. The logical architecture of DTMF.

In such a blockchain-based trust management system, TEM evaluates each node (i.e., behaving as message senders or evaluators) based on the quality of their sending messages or rating results to obtain their global trust credits. In this process, all intermediate credits in operations (e.g., S1 in Fig. 2) are temporally stored in an isolated area (i.e., a trusted execution environment, TEE) of roadside infrastructures (e.g., base stations). Thereafter, all participating nodes (i.e., evaluators in S1) have incentives based on IM (i.e., S2 in Fig. 2). Finally, the global trust credits and incentive results are maintained in a blockchain through the CM (i.e., S3 in Fig. 2). These credits are then publicly released to any node to retrieve any time without being changed again. Finally, the following sub-sections will demonstrate the details of the three models.

### *A. Trust Evaluation Model*

*Trust evaluation model (TEM)* is a collection of rules that are used to quantify the reputation of each entity based on its behaviour in the systems. According to the design paradigm, TEM includes four steps: message classification, evaluator selection, message quality rating, and trust calculation, which are labelled S1-1 to S1-4 in Fig. 2, respectively. To help readers clearly understand the following algorithms, all related variables are described in Table I.

Before starting the calculation of node trust (i.e., S1-4), the trust evaluation model must first be subjected to the following preprocessing steps:

S1-1: **Message classification (Algorithm 1)** Once a new round of trust evaluation starts (i.e., $TimeSlotActive$ is true), each base station collects messages during the past time segment in its coverage, so all the messages will eventually be collected by the system. Next, the base station groups these messages $m_{j,i,k}$s into

TABLE I
DEFINITION OF MODEL PARAMETERS

| Notations | Descriptions |
|---|---|
| $r_k, R$ | A base station $k$, and a base station set. |
| $v_i, V, V_k$ | A vehicle node $i$, the entire vehicle set, and a registered vehicle set at base station $r_k$. |
| $m_{j,i,k}, M_{i,k}$ | The $j^{th}$ message sent by vehicle $v_i$ within the coverage of base station $k$, and a message set of vehicle $v_i$. |
| $E_{j,i,k}$ | A rating result set for the message $m_{j,i,k}$. |
| $e^z_{j,i,k}$ | The $z^{th}$ rating result of the message $m_{j,i,k}$ in the set $E_{j,i,k}$. |
| $Q_{i,k}$ | A rating result set for all messages sent by vehicle $i$ at base station $k$. |
| $p(v_i)$ | An individual trust credit based on message sending of node $v_i$, referring to a specific rating result of it. |
| $P_{i,k}$ | A trust set for each nodes for message rating, to record a set of trust credits based on message rating. |
| $\tau_{mr}(v_i), \tau_{ms}(v_i)$ | The trust credits calculated on message rating activities and quality of message sending, respectively. |
| $\tau(v_i)$ | The global trust credit of node $v_i$. |
| $msgLt_{i,k}$ | A message list of vehicle node $i$ at base station $k$. |
| $atbLt_{n,k}$ | A message list of attribute type $n$ at base station $k$. |
| $idxLt_{t,k}$ | A list of index in time-cycle $t$ at base station $k$, which includes every message's attributed ID and message amount. |
| $evaLt_{t,k}$ | A list of selected evaluators for $idxLt_{t,k}$, including each attributed ID and a set of associated evaluators. |
| $msgRtLt_{j,i,k}$ | A list of rating results (i.e., a 2-tuple: $< m_{j,i,k}, e^z_{j,i,k} >$) for each message $m_{j,i,k}$. |
| $mCdtLt_k$ | A list of final rating results for all messages at base station $k$. |
| $msCdtLt_k$ | A list of overall trust credits based on message sending at base station $k$. |
| $mrCdtLt_k$ | A list of overall trust credits based on message rating at base station $k$. |
| $ndGTrsLt_k$ | A list of global trust degree of all nodes at base station $k$. |
| $vadLt_t$ | A list of qualified validators randomly selected at a $r_k$ in a given time-cycle $t$. |
| $incEvaLt_k$ | A local incentive credit list of all message evaluators at base station $k$. |
| $incVadLt_k$ | A local incentive credit list of all transaction validators at base station $k$. |
| $incLt_k$ | The incentive list of all vehicle nodes at base station $k$. |

**Algorithm 1:** Message Classification on Attribute

```
Input: $V$, $R$, $M$.
Output: $msgLt_{i,k}$, $atbLt_{n,k}$, $idxLt_{t,k}$.
Step 1: Message classification.
$TimeSlotActive = True$ ;
while $TimeSlotActive$ do
    foreach $m_{j,i,k}$ at $r_k$ do
        if $m_{j,i,k} \in G_{n,k}$ then
            Add $m_{j,i,k}$ in $atbLt_{n,k}$;
        Add the current message in $msgLt_{i,k}$ ;
        if Timeout then
            $TimeSlotActive = False$;
            Stop classifying messages;
Step 2: Index list generation.
foreach $atbLt_{n,k}$ do
    Count the number of messages: $\#m_k$ ;
    Add the current attribute ID ($Attrb_{ID}$) and associated $\#m_k$ in $idxLt_{t,k}$;
Send $idxLt_{t,k}$s to all other $r_k$s;
```

different lists ($atbLt_{n,k}$s) based on their corresponding attributes $G_{n,k}$s. Then, for each $atbLt_{n,k}$, generates an index list $idxLt_{t,k}$ that contains an attribute ID and the number of messages in $atbLt_{n,k}$. Finally, $r_k$ broadcasts $idxLt_{t,k}$ to other base stations for message evaluator selection.

S1-2: **Message evaluator selection (Algorithm 2).** Based on each message group's index $idxLt_{t,k}$, a number of evaluation candidates can be selected by calculating the similarity between the attributes of the message group (i.e., $Attrb_i$) and the vehicle node (i.e., $Attrb_{v_i}$) through Jaccard similarity. The appropriate candidate nodes are selected into a list $evaLt_{t,k}$. Then, evaluation requests are sent to candidates for their acceptance of the message evaluation until a sufficient number of evaluators (i.e., $Cout(v_i) < \xi$) are confirmed.

S1-3: **Message quality rating (Algorithm 3).** First, according to each message group $atbLt_{n,k}$ as well as its corresponding evaluator set $evaLt_{t,k}$, each message will be rated by all nodes in the associated node set. Then, a rating result list $msgRtLt_{j,i,k}$ is obtained for each message and returns it. Second, each $msgRtLt_{j,i,k}$ is converted into a new set $E_{j,i,k}$ that contains all the rating results of a message $m_{j,i,k}$. It then calculates the global rating result $\phi(m_{j,i,k})$ with Bayesian inference based on $E_{j,i,k}$ and temporally maintains the result in the list $mCdtLt_k$.

After implementing the above steps, all the messages have been evaluated and given the final global rating results that are kept in $mCdtLt_k$. In the next step, i.e., S1-4, the trust degree of the nodes will be determined by their two distinguished roles: the role acting as message senders based on the overall rating results (i.e., $mCdtLt_k$) of their sent messages, or the role acting as message evaluators based on the message rating results (i.e., $msgRtLt_{j,i,k}$) of their evaluated messages. Finally, the trust evaluation model yields a global trust credit for each node by considering both roles.

Based on the preprocessing steps, Algorithm 4 formally defines the procedure for the node trust evaluation model (i.e., S1-4 in Fig. 2). In Algorithm 4, node trust credits are evaluated by three steps:

S1-4-1: **Trust evaluation based on message rating.** Step 1-4-1 generates a trust credit based on the rating activities of each $v_i$. The step first calculates the relative difference and absolute difference between each $v_i$'s rating value $e_{j,i,k}$ and the average rating result $\phi(m_{j,i,k})$ of a message, and with such differences, it then obtains a credit $p(v_i) \in [0, 1]$ that is stored

**Algorithm 2:** Message Evaluator Selection

```
Input: $idxLt_{t,k}$, $V_k$.
Output: $evaLt_{t,k}$.
Step 1: Selection by similarity calculation.
foreach $idxLt_{t,k}$ do
    foreach $v_i \in V_k$ do
        if $Jaccard(Attrb_i, Attrb_{v_i}) > \theta$ then
            Add $v_i$ in a set $S_n$ ;
        Add $Attrb_i$ and its associated $S_n$ in $evaLt_{t,k}$ ;
Step 2: Request $v_i$ for an evaluating mission.
foreach $S_n \in evaLt_{t,k}$ do
    foreach $v_i \in S_n$ do
        if $v_i$ rejects evaluating mission then
            Remove $v_i$ from $S_n$ ;
    Update $evaLt_{t,k}$ ;
foreach $S_n \in evaLt_{t,k}$ do
    if $Cout(v_i) < \xi$ then
        Request more $v_i$s for evaluation.
```

**Algorithm 3:** Message Quality Rating

```
Input: $atbLt_{n,k}$, $evaLt_{t,k}$.
Output: $mCdtLt_k$.
Step 1: Rating each message on each evaluator.
foreach $atbLt_{n,k}$ at $r_k$ do
    foreach $m_{j,i,k} \in atbLt_{n,k}$ do
        foreach $v_i \in S_n$ corresponding to $atbLt_{n,k}$ do
            $V_i$ rates $m_{j,i,k}$ with $e^z_{j,i,k} \in [0, 1.0]$ ;
            Add the current rating result $< v_i, e^z_{j,i,k} >$ in
              $msgRtLt_{j,i,k}$ of the current $m_{j,i,k}$ ;
        Return $msgRtLt_{j,i,k}$ of the current $m_{j,i,k}$ ;
    Return a set of $msgRtLt_{j,i,k}$s of the $atbLt_{n,k}$;
Return all sets of $msgRtLt_{j,i,k}$s corresponding to all
  $atbLt_{n,k}$s at $r_k$.
Step 2: Getting global rating result of messages.
foreach $msgRtLt_{j,i,k}$ do
    foreach Rating result $< v_i, e^z_{j,i,k} >$ of $m_{j,i,k}$ do
        Add $e^z_{j,i,k}$ in a new set $E_{j,i,k}$ for $m_{j,i,k}$ ;
    Return $E_{j,i,k}$ ;
    Calculate the global rating result $\phi(m_{j,i,k})$ of message
      $m_{j,i,k}$ based on $E_{j,i,k}$ through a Baysian inference. ;
    Add $< m_{j,i,k}, \phi(m_{j,i,k}) >$ in $mCdtLt_k$ ;
Return $mCdtLt_k$.
```

in a list $P_{i,k}$ of $v_i$. Finally, the overall credit of $v_i$ based on its rating behaviours is calculated through Bayesian inference, which is $\tau_{mr}(v_i)$.

S1-4-2: **Trust evaluation based on message sending.** Step 1-4-2 generates a trusted credit based on the quality of messages sent by $v_i$. In this step, all the messages and their associated rating credits are sorted to a group of sets $Q_{i,k}$s corresponding to their hosting nodes (i.e., $v_i$s), respectively. Thereafter, an overall trust credit of $v_i$ based on the quality of the messages it sent is obtained via Bayesian inference calculation, which is $\tau_{ms}(v_i)$.

**Algorithm 4:** Node Trust Evaluation Algorithm Based on RSUs(With TEEs) and Rating Nodes

```
Input: $mCdtLt_k$, all $msgRtLt_{j,i,k}$s.
Output: $ndGTrsLt_k$.
Step 1: Trust evaluation on message rating.
foreach $msgRtLt_{j,i,k}$ do
    foreach $< v_i, e^z_{j,i,k} >$ do
        Calculate the relative difference and absolute
          difference between each $e_{j,i,k}$ and the average
          $\phi(m_{j,i,k})$;
        Obtain the trust credit of a rater $v_i$, i.e., $\in [0, 1]$,
          on this rating activity;
        Add the $p(v_i)$ to $v_i$'s corresponding $P_{i,k}$ ;
foreach $v_i$ do
    Calculate $v_i$'s trust credit $\tau_{ms}$ with its $P_{i,k}$ through
      Baysian inference;
    Add current $v_i$ and its $\tau_{ms}(v_i)$ in $mrCdtLt_k$ ;
Step 2: Trust evaluation on message quality.
foreach $m_{j,i,k} \in mCdtLt_k$ do
    if $m_{j,i,k} \in M_{i,k}$ then
        Add $m_{j,i,k}$ and its rating credit in the set $Q_{i,k}$
          being associated with $v_i$;
foreach $v_i$ do
    Calculate the $v_i$'s trust credit $\tau_{ms}(v_i)$ with its $Q_{i,k}$
      through Baysian inference ;
    Add current $v_i$ and its $\tau_{ms}(v_i)$ in $msCdtLt_k$ ;
Step 3: Global trust evaluation.
foreach $v_i$ at $r_k$ do
    Obtain a global trust credit of $v_i$ from:
    $\tau(v_i) = \omega_{mr} \cdot \tau_{mr}(v_i) + \omega_{ms} \cdot \tau_{ms}(v_i) + \omega_{his} \cdot \tau'(v_i)$,
      $\omega_{mr} + \omega_{ms} + \omega_{his} = 1$;
    Update $ndGTrsLt_k$ with $\tau(v_i)$ ;
Return $ndGTrsLt_k$;
```

S1-4-3: **Global trust evaluation.** In this step, a global trust credit of a node is obtained based on its rating behaviours (i.e., S1-4-1), the quality of messages it sent (i.e., S1-4-2), and its history credit $\tau'(v_i)$. The global credit can be acquired from the weighted average of $\tau_{mr}(v_i)$, $\tau_{ms}(v_i)$ and history trust credit $\tau'(v_i)$, which is $\tau(v_i)$ saved in list $ndGTrsLt_k$.

Since the global trust credits are solved in Algorithm 4, they will be temporally stored in the TEE-based storage embedded in each roadside infrastructure. Once the time-cycle terminates, the global trust credits will be updated to the blockchain by running a consensus model, which is detailed in the next sub-section.

### B. Incentive Model

To guarantee sustainable trust evaluation services, an incentive mechanism must be designed to reward the nodes that

**Algorithm 5:** Node Incentive Algorithm Based on RSUs(With TEEs)

```
Input: $P_{i,k}$, $\upsilon adLt_t$.
Output: Blockchain Update.
Step 1: Incentive for trust evaluators.
foreach $P_{i,k}$ do
    foreach $p(\upsilon_i) \in P_{i,k}$ do
        TEE calculates $\alpha_{i,k} = \alpha'_{i,k} \pm p(\upsilon_i) \cdot \Phi$;
        /*$\Phi$ is a unit of reward credit.*/
    TEE updates $incEvaLt_k$ with $\alpha_{i,k}$; /*$\alpha_{i,k}$ is $\upsilon_i$'s local
     evaluation rewards at base-station $k$.*/
Step 2: Incentive for transaction validators.
foreach $\upsilon adLt_t$ do
    foreach $\upsilon_i \in \upsilon adLt_t$ do
        TEE calculates $\beta_{i,t} = \beta_{i,t} \pm \Phi$;
    TEE updates $incVadLt_t$ with $\beta_{i,t}$; /*$\beta_{i,t}$ is $\upsilon_i$'s local
     validation rewards during time cycle $t$.*/
Step 3: Overall incentive for each node.
foreach $incEvaLt_k$ do
    foreach $\alpha_{i,k} \in incEvaLt_k$ do
        $\alpha_i = \alpha_i + \alpha_{i,k}$;
        /*$\alpha_i$ is $\upsilon_i$'s overall evaluation rewards.*/
foreach $incVadLt_t$ do
    foreach $\beta_{i,t} \in incVadLt_t$ do
        $\beta_i = \beta_i + \beta_{i,k}$;
        /*$\beta_{i,t}$ is $\upsilon_i$'s overall validation rewards.*/
Overall credits of $\upsilon_i$: $\sigma_i = \alpha_i + \beta_i$;
/*$\sigma_i$ is $\upsilon_i$'s global rewards in the system.*/
TEE updates $incLt$ with each $\sigma_i$, and writes to the
 blockchain via Algorithm 6.
```

make positive contributions to services. Algorithm 5 defines a set of incentive criterion of nodes, including trust evaluators and transaction validators that are implemented in the TEE of each base station. The algorithm consists of three key steps as follows:

S2-1: **Incentive for trust evaluators.** is conducted based on the trust credit $p(\upsilon_i)$ that is used as a factor to adjust the incentive intensity, which is achieved by multiplying $p(\upsilon_i)$ by a unit of reward credit $\Phi'$. The total incentive reward $\alpha_{i,k}$ of an evaluator is obtained by adding up a reward associated with an evaluating mission, and the add-on reward value is either positive for correct nodes or negative due to malicious node behaviours. Finally, each $\alpha_{i,k}$ is stored in a list named $incEvaLt_k$.

S2-2: **Incentive for transaction validators.** is similar to that used for evaluators, but the rewards are based on the number of transaction-validating missions completed by each validator. The total reward $\beta_{i,t}$ is calculated by adding up or deducting a reward unit $\Phi'$ depending on the node behaviours during each validating mission, and then the reward value is saved in the list $incVadLt_t$.

S2-3: **The overall incentive reward for each node.** is accumulated from mission rewards obtained by a node based on its participation in the trust evaluation, transaction validation or both. The overall reward value is represented as $\sigma_i = \alpha_i + \beta_i$, which is saved in list $incLt$.

According to the design of DTMS, the trust evaluation algorithm first calculates trust credits of nodes based on their evaluating behaviours. Then, the incentive algorithm is applied to assess the node reward or penalty based on the amount and quality of the missions achieved by the nodes. It is worth mentioning that the calculation of incentive rewards is implemented and secured by TEEs integrated with base stations in terms of Algorithm 5. Finally, both trust credits and reward values are updated to the blockchain by running the consensus protocol, namely, Algorithm 6, which is introduced in the following section. Hence, the distributed structure of Algorithms 4 and 5 guarantees the decentralization of DTMS, and Algorithm 6 ensures the traceability and irreversibility of trust credits and reward values.

### *C. Consensus Model*

DTMS is designed with a blockchain system to preserve the trust credit and incentive reward of each node. Each update of the trust credit or incentive reward is specified as a transaction, and a group of transactions forms a block. The consensus model, i.e., Algorithm 6, is designed to carry out such block generation and validation during a consensus process before appending to the blockchain. The detailed consensus procedure is introduced as follows:

**Step 1**: Select validators randomly from a group of qualified candidates.

1). Each base station $r_k$ (TEE) sorts its registered nodes in the current round and finds all qualified nodes with trust credits greater than a given value $\kappa$.

2). If the qualified nodes agree to participate in the validation, then they will be reserved in a candidate list held by the TEE, i.e., $\upsilon adCanLt_t$, which is shared with other TEEs.

3). The TEE randomly selects a group of nodes from all candidate lists to be validators of the current consensus round, and it reserves them in list $\upsilon adLt_t$.

In this step, all operations are secured by TEEs, and validators are randomly selected for both fairness and efficiency. As only qualified nodes can be selected as potential validators, it is not necessary to include abundant validators in the consensus, which does not compromise security but gains higher efficiency.

**Step 2**: Validate transactions through selected validators.

1). TEE broadcasts a block to all selected validators directly or via other TEES;

2). Validators verify each transaction by recalculating its value based on Algorithms 4 and 5, and then sign the block as either "VALID" or "INVALID";

3). TEE must collect "VALID" validation messages from over $2/3$ of all validators to continue the following consensus step.

Multi-signature technology is leveraged by validators to generate efficient and aggregated signatures, which can reduce the message complexity of this step. Moreover, the validation is running parallelly in the system, as the step is independently organized by each base station (TEE). Such parallel design can achieve better system scalability and higher consensus performance.

**Step 3**: Consensus for block validation and blockchain update.

1). TEE broadcasts its block to other TEEs to verify the multi-signature.
2). TEE verifies all received blocks' multi-signature to confirm the block validity and then sends the signed confirmation message to the block sender;
3). After receiving confirmations from over $1/2$ of all TEEs, the block can be committed to being ready for the update.
4). TEE takes turns updating its new block to the blockchain.

In the above consensus step, validated blocks must be verified by more than half of the TEEs to ensure that all the signatures are valid. This is the final consensus among TEEs to commit the validity of a block. As TEE is assumed to be a secure hardware environment, only crash faults may occur between TEEs, so the consensus can be agreed by only over $1/2$ of all TEEs. However, for transaction validation, a valid block must be confirmed by more than $2/3$ of all validators without TEE secured hardware. Moreover, DTMS executes the consensus procedure in a two-layer manner: the transaction validation that is based on each TEE in parallel and implemented by the selected validators; the block verification and consensus that are conducted on a top layer by all TEEs. Such a layered architecture design achieves the parallelism of the consensus protocol, which improves the system scalability and efficiency.

## V. Security Analysis

Blockchain technology is applied to the DTMS due to its reliability and security in data storage. Furthermore, DTMS also implements a new trust-based consensus protocol to improve the efficiency of reaching agreements but still guarantee security. This section provides the required security analysis for the DTMS.

*Security* property of the consensus protocol is analysed by discussing how the protocol resists potential security risks that are illustrated in the following propositions.

*Proposition 1: The related data for trust evaluation and incentive assessment are secured from external attackers (e.g., Sybil attackers).*

*Proof:* First, each node must ne registered in the system with its unique vehicle identity. Each node, including TEE, has a unique key pair for encryption and signature, and the public keys are only allowed to be shared among registered vehicle nodes and TEEs. All data transferred between nodes and base stations are encrypted. For example, the lists of message rating results must be encrypted with the corresponding TEEs' public keys before sending them to the base stations for further calculation and signed with the sender's private key. This process ensures that only TEE can decrypt these lists and retrieve the data, and only valid nodes can participate in evaluating activities with a unique identity.

**Algorithm 6:** Consensus Protocol Based on RSUs(With TEEs) and Validating Nodes

```
Input: $ndGTrsLt_k$.
Output: Blockchain Update.
Step 1: Validator selection at each $r_k$.
/*Validators are randomly selected from qualified candidates.*/
while $NotSufficientValidators = true$ do
    foreach $r_k$ do
        for $v_i \in V$ do
            if $\tau'(v_i) > \kappa$ and $v_i$ ACCEPTS then
                Add $v_i$ in $vadCanLt_t$;
        Send $vadCanLt_t$ to other $r_k$s;
    foreach $r_k$ do
        TEE randomly selects a group of $v_i$s from received $vadCanLt_t$s as validators;
        Update $vadLt_t$ with selected validators;
    if $\#vadLt_t \geq n$ at each $r_k$ then
        NotSufficientValidators = false;
Step 2: Transaction validation.
/*$Tx$ is a transaction recording vehicle identity information and its trust credit or incentive reward, i.e., $< v_i, \tau(v_i) >$ or $< v_i, \sigma(v_i) >$, and other related information.*/
foreach $r_k$ do
    TEE broadcasts a block (i.e., a group of $Tx$s) to all validators in its $vadLt_t$;
    Validators verify each $Tx$ in the block, mark the block as "VALID" or "INVALID" with multi-signatures, and retrun the signed block to TEE;
    TEE counts up the number of signatures (i.e., $\#SIGs$) that mark the block "VALID";
    if $\#SIGs > \frac{2}{3} \cdot \#vadLt_t$ then
        Confirm the block to be ready for updating;
    else
        Restart validation after excluing invalid $Tx$s;
Step 4: Blockchain update.
foreach $r_k$ do
    TEE signs its own confirmed block and boradcast to other TEEs;
    TEE also verifies the signagures of all received confirmed blocks, i.e., $Bk$s;
    if $Bk$ is valid then
        Sign "VALID" and return to its sender;
    else
        Sign "INVALID" and return to its sender;
    if TEE recevies "VALID"s from over $1/2$ of all TEEs then
        Commit the block and take truns updating the blockchain;
```

Furthermore, while conducting trust evaluation and incentive assessment based on Algorithms 4 and 5, TEE is employed in each base station for secure and reliable computation. Only the final trust credits and incentive values can be broadcasted as transactions over the network. Moreover, the digital signature is adopted to ensure data integrity. Hence, with the public key infrastructure and digital signatures, the confidentiality of the data is guaranteed in DTMS. □

*Proposition 2: Any adversaries (e.g., malicious message raters or consensus validators) have no chance to alter the truth of an event by spreading fake messages (e.g., message spoofing attack) or creating a rating with bias (e.g., bad mouthing and ballot stuffing attacks).*

*Proof:* The node selection based on trust credit ensures that most selected nodes (usually $> 2/3$ of all based on classic Byzantine fault-tolerant cases) for message rating or trust evaluation are not adversaries. Therefore, even if some malicious nodes may send fake messages, these correct nodes can generate a rational rating without bias, which can reduce the negative effect of fake messages.

In a worse situation, if some malicious nodes generate unfair ratings, and such ratings can be submitted to trust evaluating operations, those malicious nodes will eventually be penalized by the incentive model. Like the assumption that over two-thirds of nodes are correct, the eventual rating results should be rational and trustful. Thus, the rating values created by malicious nodes must have an obvious deviation from the average. The incentive model can then distinguish such a deviation and generate a penalty. Hence, the trust and incentive mechanism can guarantee the trustworthiness of participating nodes and the fairness rating. □

*Proposition 3: The node trust evaluation and incentive assessment are still secure in the situation of some compromised/malicious base stations or even over $2/3$ malicious base stations .*

*Proof:* In the layer of base stations, external attackers may compromise some base stations, or the base stations may behave as malicious nodes for selfish intents. In the compromised base stations, attackers may tamper with the message rating results after decryption. However, they cannot forge a valid signature without the rating nodes' secret keys, as all rating results have been signed with raters' secret keys before being sent to the base stations. Hence, the attackers will eventually fail to pass the verification taken on other base stations while conducting trust evaluating operations. Conversely, for the malicious base stations, the adoption of TEE provides a secure environment independent of the distrustful hardware and applications for reliable trust and incentive computation, regardless of malicious base stations. Thus, malicious base stations have no chance to interfere in those TEE-secured operations as long as the base stations are not physically damaged.

Finally, supposing TEE may generate incorrect trust credit or incentive values in transactions, with the consensus algorithm (i.e., Algorithm 6), validators must recalculate such credits and values based on the publicly released Algorithms 4 and 5 for conducting validation and then achieve an agreement with other validators regarding the validity of transactions. Even if some incorrect transactions may be generated from TEEs, they will be detected through validation. As a result, a blockchain is adopted to preserve trust credits and incentive values. □

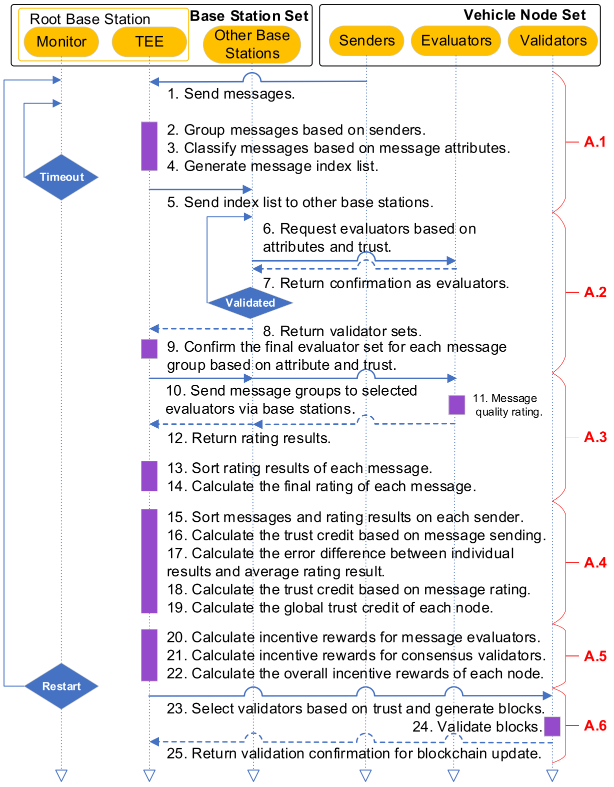

Root Base Station
Monitor
TEE
Base Station Set
Other Base Stations
Vehicle Node Set
Senders
Evaluators
Validators
Timeout
1. Send messages.
2. Group messages based on senders.
3. Classify messages based on message attributes.
4. Generate message index list.
5. Send index list to other base stations.
6. Request evaluators based on attributes and trust.
7. Return confirmation as evaluators.
Validated
8. Return validator sets.
9. Confirm the final evaluator set for each message group based on attribute and trust.
10. Send message groups to selected evaluators via base stations.
11. Message quality rating.
12. Return rating results.
13. Sort rating results of each message.
14. Calculate the final rating of each message.
15. Sort messages and rating results on each sender.
16. Calculate the trust credit based on message sending.
17. Calculate the error difference between individual results and average rating result.
18. Calculate the trust credit based on message rating.
19. Calculate the global trust credit of each node.
20. Calculate incentive rewards for message evaluators.
21. Calculate incentive rewards for consensus validators.
22. Calculate the overall incentive rewards of each node.
Restart
23. Select validators based on trust and generate blocks.
24. Validate blocks.
25. Return validation confirmation for blockchain update.
A.1
A.2
A.3
A.4
A.5
A.6

Fig. 3. Activity flow of BTMF on the scenario of VSNs.

## VI. Performance Evaluation

The security properties of the proposed framework have been analysed, and the next section will explore the performance issues in the scenario of VSNs. For performance evaluation, the section initially defines a performance model based on a series of activities when implementing DTMS on a vehicular network; formal analysis is conducted by adopting a novel formal approach (i.e., Performance Evaluation Process Algebra, PEPA) due to its advantages in efficient modeling and feature analysis. Additionally, to confirm the superior performance of DTMS, practical experiments are also implemented by building a testbed and comparison with other similar schemes.

### *A. Performance Model Scenario*

According to the design of DTMS and its practical application scene, the performance model is created containing two types of entities: *base stations* and *vehicle nodes*. The base stations undertake most operations of the designed scheme, while the vehicle nodes can behave as message senders, message quality evaluators or blockchain validators. Fig. 3 shows a sequence of activities in the deployment of DTMS on VSNs.

In Fig. 3, every base station includes two parts: a monitor managing the start and end of each round, and a trusted execution environment (TEE) providing secure hardware; the vehicle nodes include three types of roles that are message senders, evaluators and consensus validators.

*Algorithm 1:* The activity flow starts from collecting messages (Step 1), and the root base station initially groups all the received messages by senders (Step 2) and classifies these messages in terms of various attributes to an obtain attribute-based message classifications,i.e., $atbLt_{n,k}$ (Step 3); thereafter, an index (i.e., $idxLt_{t,k}$) is generated, which consists of all the classification information at the base station $k$ (Step 4). Finally, the root station broadcasts the index to other base stations for message evaluation (Step 5).

*Algorithm 2:* Before starting the message evaluation, a group of evaluators must be selected based on their attributes and trust. Thus, the base stations send requests to their connected evaluation candidates based on the required attribute type and trust credit (Step 6). Then, if the nodes are willing to join the evaluation, they will confirm the requests to become evaluators and notify the base stations (Step 7). Once the base stations verify the qualification and quantity of evaluators, they will send the selected evaluators to the root base station (Step 8) on which these evaluators will eventually be confirmed in $evaLt_{t,k}$ (Step 9).

*Algorithm 3:* Once the root base station confirms the selected evaluators, it sends each message group to its corresponding evaluators (Step 10) at which the messages are rated (Step 11) for quality, and the rating results are then returned (Step 12) to the root base station. The final rating result of a message is calculated at the TEE of the root base station (Step 13-14) and stored in $mCdtLt_k$.

*Algorithm 4:* Node trust evaluation is conducted after obtaining all final message rating results based on each sending node. The base station performs a series of calculations in its TEE to obtain trust credits based on the message rating and sending records (Steps 15-18). The global trust credit $\tau(v_i)$ of each node is calculated based on the assumed roles by running a weighted average calculation (Step 19). The global trust credit is temporally stored in $ndGTrsLt_k$ until being updated to the blockchain.

*Algorithm 5:* The incentive algorithm calculates the rewards of nodes based on their workload and quality, such as the number of evaluated messages and the quality of the rating results (Step 20). The incentive rewards also include the work of consensus validation in the last consensus round (Step 21). Hence, the overall incentive rewards are obtained and temporally saved in $incLt_k$ (Step 22).

*Algorithm 6:* Finally, the consensus algorithm is implemented to update newly generated node trust credits and incentive rewards to the blockchain through the base station's TEE, which leverages a group of selected validators for block validation (Steps 23-25).

### B. Formal Modeling and Analysis

According to the performance modeling scenario, we initially leverage a formal analysis method, i.e., performance evaluation process algebra (PEPA), which is a high-level model specification language, defined by Hillston [31]. PEPA is defined with a random duration following an exponential distribution. Due to the memoryless property of the exponential distribution, the stochastic process indicated by PEPA has the Markov property. Hence, the underlying stochastic process is a continuous-time Markov chain.

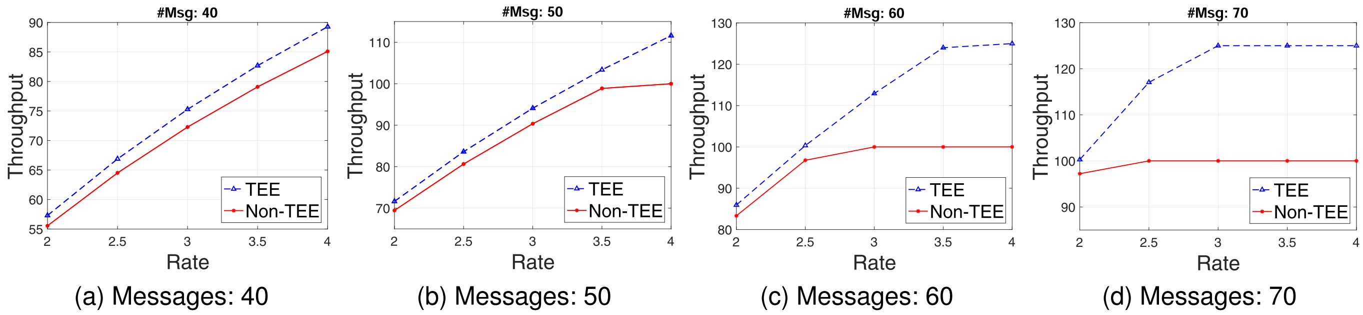
PEPA has a dramatic advantage: efficient and accurate model creation and analysis in comparison to a time-consuming modeling process based on a simulation or practical experiment. PEPA's key benefits can be highlighted as follows. *Formality*: PEPA language has structured operational semantics and provides a formal interpretation for all expressions; *compositionality*: the compositional nature provides the ability to model a system as the interaction of subsystems; *abstraction*: PEPA can construct complex models from detailed system components, disregarding details when it is appropriate to do so.

As the simulation and experiment have drawbacks of time cost in building models and measuring performance, it would be efficient to leverage such a formal approach to analyse the initial design. PEPA can generate efficient and accurate analyses because of a novel approach called *fluid flow approximation* [32], which can generate a continuous state space approximation with evolution governed by a set of ordinary differential equations (ODEs). The performance matrices can be directly obtained by solving such ODEs. The details of PEPA language and fluid flow approximation can be obtained in Ref. [31], [32].

*Performance Criterion:* The performance of DTMS is evaluated by comparison to a trust evaluation scheme without using TEE and a blockchain consensus scheme using PoW, respectively. The non-TEE trust evaluation scheme implements trust evaluation operations under an encryption/decryption scheme to secure all the data processed, while DTMS utilizes TEE to provide a secure space at each based station, which can save the cost of using an encryption/decryption scheme. Figs. 4, 5 and 6 represent the performance benchmark between DTMS and the non-TEE trust evaluation scheme. Conversely, we also compared the efficiency of data management on DTMS, which is based on a blockchain to preserve both trust credits and incentive rewards. DTMS' blockchain achieves consensus by a consortium (i.e., TEEs and a group of selected validators based on their trust credit) rather than allowing all nodes to participate in the consensus process such as PoW. Thus, we analyse DTMS' consensus performance by comparison to a general non-consortium consensus, as represented in Figs. 7, 8, 9 and 10.

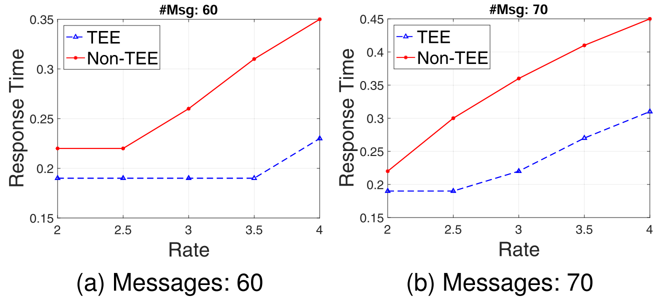
*Analysis Configuration:* PEPA is used to define the activities of DTMS based on Fig. 3 and implemented in Eclipse with a PEPA plugin that can support the specification of DTMS' formal model and run the fluid flow analysis. The analysis includes *throughput*, which is the number of complete trust evaluation rounds during a specified time interval, and *response time*, which is the average time spent completing a round of trust evaluation. The total number of nodes varies from 40 to 70 with a step size of 10, and we assume that each base station connects 10 nodes. The time unit in the formal analysis is a default time unit in the analyser.

*DTMS vs. Non-TEE Scheme:* Fig. 4 depicts the average system throughput in the process of message evaluation.



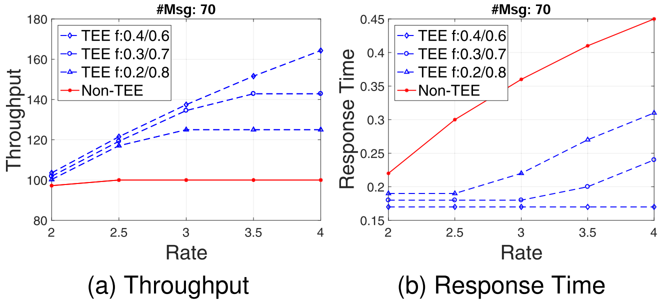
(a) Messages: 40 (b) Messages: 50 (c) Messages: 60 (d) Messages: 70

Fig. 4. Average throughput of system on message evaluation with TEE or without TEE.



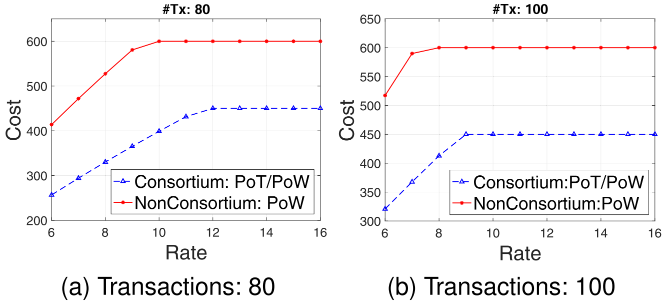
(a) Messages: 60 (b) Messages: 70

Fig. 5. Average response time on message evaluation with TEE or without TEE.



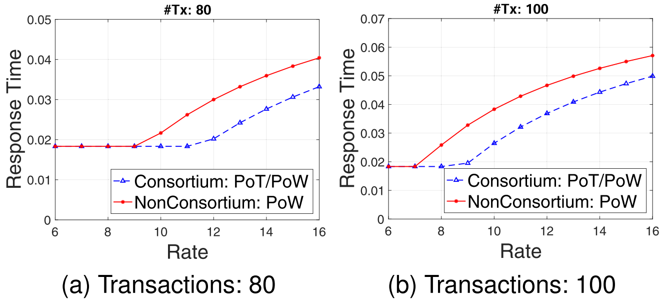
(a) Throughput (b) Response Time

Fig. 6. Average throughput and response time on message evaluation varying against factors $f$.



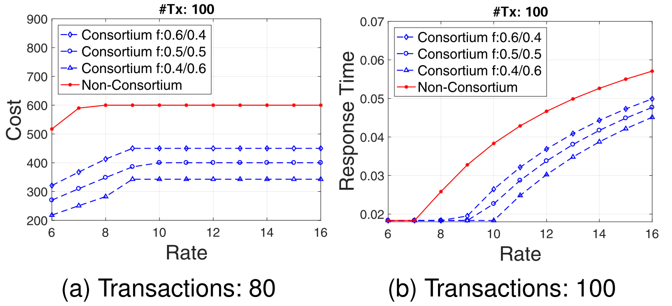
(a) Transactions: 80 (b) Transactions: 100

Fig. 7. Average cost on transaction validation with two consensus protocols.



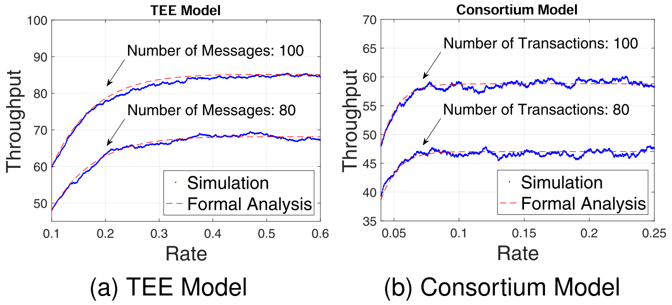
(a) Transactions: 80 (b) Transactions: 100

Fig. 8. Average response time on transaction validation with two consensus protocols.



(a) Transactions: 80 (b) Transactions: 100

Fig. 9. Average cost and response time on transaction validation varying against factors $f$.



(a) TEE Model (b) Consortium Model

Fig. 10. Simulation-based throughput validation on TEE and consortium models.

In the experiments, the number of messages in the process is set from 40 to 70 with a fixed step size of 10, and the associated figures are shown in Figs. (a)-(d), respectively. According to the set of figures, the TEE-based framework (i.e., DTMS) generates higher throughput with the growing message arrival rate. Finally, the throughputs reach stationary values that are approximately 125 for the TEE-based DTMS and 100 for the non-TEE framework. Similarly, Fig. 5 shows the average response time for the message evaluation under the two cases of 60 and 70 messages in the system, respectively. Figs. (a) and (b) both present the trend in which the TEE scheme yields a faster response than the non-TEE framework, as the blue lines are always located below the red lines in the figures. Finally, we also analyse the performance of DTMS

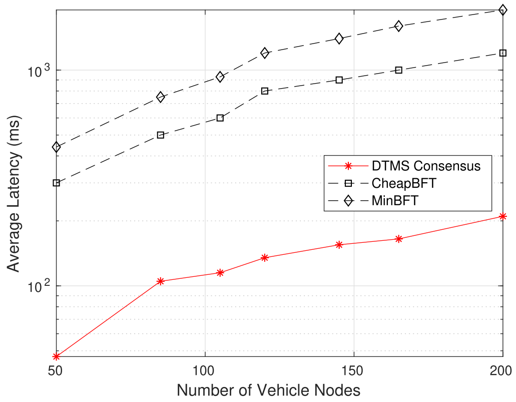


Fig. 11. Average throughput of DTMS.

varying against a *factor*, which indicates the proportion of transactions that are stored temporally in the TEE. As shown in Fig. 6, $f$ : 0.4/0.6 means that 40% transactions are preserved in the TEE before updating to the blockchain, while 60% are still running in the system. Concerning the plots of Fig. 6 (a), the non-TEE scheme has the least throughput, whereas the throughput of the TEE-based DTMS gradually increases with the factor varying from 0.2/0.8 to 0.4/0.6. Fig. 6 (b) indicates the change in average response time, which is reduced with an increasing factor from 0.2/0.8 to 0.4/0.6. Consequently, from Figs. 4, 5, and 6, the TEE-based DTMS always has better performance (i.e., greater throughput and reduced response time) than the non-TEE framework, and its performance is constantly improved with a progressively larger factor.

*DTMS vs. Non-Consortium Consensus:* Figs. 7 and 8 present the average cost and average response time for the process of validating transactions, respectively. In Fig. 7, the average cost (i.e., computing and networking cost) is obtained by multiplying the number of validated transactions with an average cost unit. The proposed DTMS with a consortium network yields a lower cost than the non-consortium framework. Moreover, when the system reaches its full capacity, the costs of consortium and non-consortium schemes tend towards equilibrium with values of approximately 450 and 600, respectively. In Fig. 8, the consortium-based DTMS generates a faster response in the consensus process, and the two lines begin to grow until reaching full capacity. Finally, Fig. 9 reveals the altered performance of two schemes against a varying factor that defines the proportion of selected transaction validators (e.g., $f$ : 0.6/0.4 means 60% nodes are selected as validators, while 40% remain idle). Figs. 9 (a & b) show that both the average cost and average response time rise gradually with an increasing factor from 0.4/0.6 to 0.6/0.4. Consequently, from Figs. 7, 8, and 9, the consortium-based DTMS yields a lower cost and faster response, in contrast to the non-consortium framework; furthermore, by preserving the reliability in a certain domain, the smaller the factor (i.e., the lower the proportion of validators), the higher is the performance.

To validate the accuracy of the formal analysis, we also use discrete event simulation to verify the analysis results from the PEPA models. Fig. 10 indicates the verification based on the TEE model and the consortium model. According to the plots, the PEPA-based formal analysis yields similar results to the simulation models, which means that the formal analysis can be considered reliable and accurate.

In conclusion, DTMS generates better performance in both the trust evaluation process using TEE environments and the consensus process designed with a selected consortium. To verify the performance of DTMS in practice, we perform an additional experimental analysis in the following section.

### C. Practical Experiments and Measurements

To verify the consensus performance of DTMS in a real-world environment, we build a testbed in Java language and deploy it on the Microsoft Azure Cloud with 8 F8s-v2 instances. Each instance includes an 8-core vCPU based on the Intel Xeon Platinum 8168 (SkyLake) processor with 16 GiB RAM and 64 GiB storage. The number of nodes applied in DTMS is set to vary from 50 to 200, and each base station is assumed to connect with 10 nodes. For the blockchain configuration, the block size is 1 MB, and the transaction size in the block is 100 Bytes.

The consensus protocol of DTMS has been designed on a novel parallel Byzantine fault-tolerant (BFT) algorithm, which can support the consensus concurrently running on each base station to improve the overall performance. As the key target of this paper is to demonstrate an overall framework, the details of the new consensus algorithm will be introduced in our subsequent research work. Here we just briefly present its performance based only on the average latency of the consensus process, as the latency is a significant criterion representing the quality of service. The practical experiment compares our consensus design with the other two classic consensus protocols, i.e., CheapBFT [30] and MinBFT [34], both of which are efficient Byzantine fault-tolerant consensus protocols and widely investigated in past years. The performance benchmark is conducted on the same Azure platform stated previously.

Fig. 11 shows the average latency of the DTMS consensus in comparison to the other two protocols. The fast consensus speed of DTMS is compared to the other two schemes. It is clear that with an increased number of nodes from 50 to 200, the average latency of the DTMS consensus only varies from 47 ms to 210 ms; however, CheapBFT increases roughly from 300 to 1200 ms, and MinBFT increases from 440 to 1900 ms. Such a large difference is due to the parallel design of the DTMS consensus protocol. Conversely, low latency means that the system can gain a higher throughput in the consensus process. Consequently, the practical experiments confirm that DTMS has an efficient consensus design, including high throughput and low latency, which can be well applied to a large-scale transportation environment.

## VII. Conclusion

This paper proposes a decentralized trust management system (DTMS) that is designed to achieve vehicle node trust evaluation and management on a blockchain-based system. We design an efficient trust evaluation algorithm that relies on a design of attribute-based evaluator selection to increase the quality of message evaluation, and an incentive design to encourage more node joining in the evaluation to ensure the sustainable service. It is worth mentioning that the trust evaluation algorithm is designed by leveraging TEE technology to provide a secure space on base stations to reduce the security cost and obtain higher performance. Additionally, blockchain technology is applied to preserve both trust and incentive data through an efficient consensus protocol, which is designed to utilize a group of trusted nodes to support consensus validation rather than conducting the consensus over the entire node set. Such a design gains performance that is 100 times superior to other classic schemes and ensures the safety of the blockchain. Hence, DTMS provides a comprehensive framework for building a blockchain-based trust management system in an intelligent transportation environment.

## References

[1] A. M. Vegni and V. Loscri, "A survey on vehicular social networks," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2397–2419, 4th Quart., 2015.

[2] Q. Yang and H. Wang, "Toward trustworthy vehicular social networks," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 42–47, Oct. 2015.

[3] T. H. Luan, X. Shen, F. Bai, and L. Sun, "Feel bored? Join verse! Engineering vehicular proximity social networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 1120–1131, Mar. 2015.

[4] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management of Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Apr. 2014.

[5] S. Li and X. Wang, "Quickest attack detection in multi-agent reputation systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 653–666, Aug. 2014.

[6] M. E. Mahmoud and X. Shen, "An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 3947–3962, Oct. 2011.

[7] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[8] C. A. Kerrache, N. Lagraa, C. T. Calafate, J.-C. Cano, and P. Manzoni, "T-VNets: A novel trust architecture for vehicular networks using the standardized messaging services of ETSI ITS," *Comput. Commun.*, vol. 93, pp. 68–83, Nov. 2016.

[9] S. Landau, "Making sense from snowden: What's significant in the NSA surveillance revelations," *IEEE Secur. Privacy*, vol. 11, no. 4, pp. 54–63, Jul. 2013.

[10] S. Landau, "Highlights from making sense of Snowden, PartII: What's significant in the NSA revelations," *IEEE Secur. Privacy*, vol. 12, no. 1, pp. 62–64, May 2014.

[11] L. Lamport, R. Shostack, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[12] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[13] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 4095–4108, Nov. 2012.

[14] C. Lai, K. Zhang, N. Cheng, H. Li, and X. Shen, "SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1559–1574, Jun. 2017.

[15] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2140–2148.

[16] D. Huang, X. Hong, and M. Gerla, "Situation-aware trust architecture for vehicular networks," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 128–135, Nov. 2010.

[17] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. 27th Conf. Comput. Commun.*, Phoenix, AX, USA, Apr. 2008, pp. 1238–1246.

[18] A. Moustafa, M. Zhang, and Q. Bai, "Trustworthy stigmergic service compositionand adaptation in decentralized environments," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 317–329, Mar. 2016.

[19] Z. Li, and C. T. Chigan, "On joint privacy and reputation assurance for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 268–273, Apr. 2014.

[20] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. H. Ngu, "CloudArmor: Supporting reputation-based trust management for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 367–380, Feb. 2016.

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

[22] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A Proof-of-Trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 429–445, May 2019.

[23] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale Internet of Things data storage and protection," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 762–771, Sep. 2019.

[24] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," 2017, *arXiv:1709.05748*. [Online]. Available: http://arxiv.org/abs/1709.05748

[25] J. Lind, O. Naor, I. Eyal, F. Kelbert, P. Pietzuch, and E. Gun Sirer, "Teechain: A secure payment network with asynchronous blockchain access," 2017, *arXiv:1707.05454*. [Online]. Available: http://arxiv.org/abs/1707.05454

[26] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.

[27] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[28] X. Chen and L. Wang, "A cloud-based trust management framework for vehicular social networks," *IEEE Access*, vol. 5, pp. 2967–2980, 2017.

[29] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.

[30] M. Blaze *et al.*, "Experience with the key note trust management system: Applications and future directions," in *Proc. 1st Int. Conf. on Trust Manage.*, 2003, pp. 284–300.

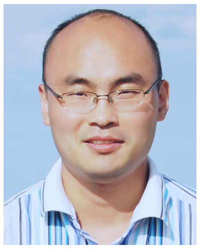
[31] J. Hillston, *A Compositional Approach to Performance Modelling*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[32] J. Hillston, "Fluid flow approximation of PEPA models," in *Proc. 2nd Int. Conf. Quant. Eval. Syst. (QEST)*, Washington DC, USA, 2005, pp. 33–43.

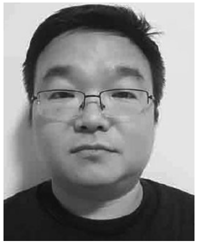
[33] R. Kapitza *et al.*, "CheapBFT: Resource-efficient byzantine fault tolerance," in *Proc. 7th ACM Eur. Conf. Comput. Syst.*, Bern, Switzerland, 2012, p. 295.

[34] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient byzantine fault-tolerance," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 16–30, Jan. 2013.

**Xiao Chen** received the M.Sc. and Ph.D. degrees in computing science from Newcastle University in 2009 and 2013, respectively. He is currently a Research Fellow with the School of Informatics, The University of Edinburgh, U.K. His research interests include performance evaluation and stochastic optimization for large-scale/distributed systems, e.g., the IoT systems, cyber-physical systems, cloud/fog systems, and blockchain systems.



**Jie Ding** received the B.S. degree in mathematical education from Yangzhou University, Yangzhou, China, in 2001, the M.S. degree in mathematical statistics from Southeast University, Nanjing, China, in 2004, and the Ph.D. degree in communication from The University of Edinburgh, U.K., in 2010. He is currently a Professor with Shanghai Maritime University. His research interests include performance modeling for communication and computer systems.



**Zhenyu Lu** received the B.Sc. degree in electricity and the M.Sc. degree in information and communication from the Nanjing Institute of Meteorology, Nanjing, China, in 1999 and 2002, respectively, and the Ph.D. degree in optics engineering from the Nanjing University of Science and Technology, Nanjing, in 2008. He was a Research Associate with the Department of Mathematics and Statistics, University of Strathclyde, Glasgow, U.K., from 2012 to 2013. He is currently a Professor with the School of AI, Nanjing University of Information Science and Technology. He has published seven international journal articles. His current research interests include neural networks, stochastic control, and artificial intelligence.