Mobile Networks and Applications (2020) 25:314–327
https://doi.org/10.1007/s11036-019-01285-9

Consortium Blockchain-Based Secure Software Defined Vehicular Network

Ning Zhao[1] · Hao Wu[1] · Xiaonan Zhao[2]

Published online: 14 June 2019

Abstract

The vehicular ad hoc network (VANET) is a promising technology that can provide Internet access services for vehicles. With the development of VANET, tremendous intelligent vehicles will emerge mass and different communication requirements. Software-defined networking (SDN) is regarded as a potential technology to enhance network performance. In recent years, a new networking paradigm called software defined vehicular networks (SDVN) has been proposed. Nevertheless, the security issues still need to be considered for SDVN, because malicious vehicles can put forward fake requirements on the control plane of SDVN, which deteriorates the network performance in a certain degree. In this paper, we associate resources allocation problem with trust value of vehicles for the first time. The trust value of vehicles can be obtained through trust management system. Considering that there are many defects in the state-in-art trust management schemes, in this paper, a decentralized trust management architecture is designed which constitutes of three layers based on consortium blockchain. A joint proof-of-stake and modified practical Byzantine fault tolerance (PoS-mPBFT) algorithm is proposed to the shorten the confirmation time, which is deployed on RSUs. Different from previous researches that focus on designing methods to evaluate trust value, we use prediction model to estimate trust value of vehicles in the next period. After calculating trust value of vehicles, it assigns more resources to those high credibility vehicles when SDN services are provided. Meanwhile, to increase the efficiency of resource allocation, we convert the multiple-path mapping problem of the virtual network into the multi-commodity flow problem, which is solved by a heuristic algorithm. The simulation results indicate that the proposed trust management architecture and heuristic algorithm could provide better safety in SDVN and shorten consensus time, meanwhile effectively abstract underlying resources to enhance network load balance and capacity.

Keywords SDN · Vehicular network · Trust management · Consortium blockchain

1 Introduction

Vehicular ad hoc networks (VANETs) plays an important role in enhancing road safety, traffic management, and network performance which is a significant cross scenario in the Smart Cities and Internet of Things [1]. With the increasing of intelligent vehicles, the existing architecture cannot satisfy the rapidly growing requirements of those vehicles. During the development of VANETs, Software-Defined Network (SDN) is a promising network paradigm to enhance the network performance and management with the advantages of dynamic network resource allocation, flexibility and programmability [2]. SDN decouples control layer and data layer and all resources of networks are processed by a logically centralized controller. Originally, SDN was designed for wired network environment, whereas in fact SDN brings new insights and high potential to improve the flexibility, programmability, efficiency, and evolvability for wireless networks [3, 4]. Therefore, the convergence of SDN and vehicular network called Software-Defined Vehicular Network (SDVN) has been a promising technology to the network structure [5, 6], and a promising network architecture for the next generation VANETs.

✉ Hao Wu
hwu@bjtu.edu.cn

Ning Zhao
ningzhao@bjtu.edu.cn

1 State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

2 School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

In the development of SDVN, in literature [5], the authors proposed using cellular network for control plane and ad hoc technology for data plane. On the control plane, vehicles put forward different resources allocation demands which will be assigned by RSU or base station. Whereas, with the exploding resources demands of vehicles, it is urgent to design an algorithm to map physical resources into different virtual nets efficiently and dynamically, i.e., physical resources allocation problem. In addition, some malicious vehicles will put forward fake resources allocation requirements which not only disrupt the normal operation in SDVN but also deteriorate the quality of service (QoS) of other vehicles, even occur to network congestion. Based on the two problems above, it is reasonable to allocate more physical resources to those high credibility vehicles. The credibility of a vehicle can be calculated through history behaviors, which is related with the validating of the road-relevant messages it reports to its neighbor vehicles. Therefore, in this paper, we focus on establishing and combining a secure and efficient resource allocation system for SDVN.

Toward securing SDVN, we focus on trust management established in SDVN. Generally speaking, there are two ways to construct the trust management systems in vehicular networks, centralized and decentralized respectively. In the centralized trust management system, all road-relevant messages are sent to a central server working in the cloud area, meanwhile, the processing and evaluation operations are also handled in this central server [7]. However, the centralized trust management system cannot always satisfy the rigorous (QoS) requirements and latency demands in vehicular networks, due to the limited network bandwidth and the server processing capability. In the traditional decentralized trust management systems, trust evaluation processes are accomplished on vehicle side: each vehicle is responsible to judge the credibility of road-relevant messages sending from neighbor vehicles and calculate the rating of senders [8]. Whereas, restricted to the limited observation sets (i.e., communication coverage) of vehicles in the decentralized scheme, the calculation results have one-sidedness. Some schemes proposed that the calculation and judgment procedures should be handled on RSU side [9], whereas, different RSUs might have different trust values at the same vehicle. Therefore, to overcome the deficiencies of the traditional trust management systems, we design a consortium blockchain-based decentralized trust management system for SDVN.

Toward enhancing flexibility and efficiency of SDVN, we focus on abstracting physical resources and map into virtual networks dynamically and efficiently. The resources allocation problem is associated with the trust value of vehicles. Then we use multi-commodity flow algorithm in SDVN to allocate physical resources. The main contributions of this paper are the following aspects:

1. A decentralized trust management architecture for SDVN based on consortium blockchain is designed. The architecture consists of three layers, which not only enables all RSUs to store and update the distributed vehicle-rating-ledger, but also ensure the privacy protection of vehicles using a trustworthy entity certification authority (CA).
2. A joint proof-of-stake and modified practical Byzantine fault tolerance (PoS-mPBFT) algorithm is proposed to enhance efficiency and security in the consensus process. Each RSU can collect, process information uploaded from vehicles and update ledger periodically.
3. Prediction model is used to estimate trust value of vehicles. We use the typical least square method to calculate the trust value of vehicles. Since each RSU holds the same ledger, the trust value calculated from different RSUs for the same vehicle is identical. The obtained trust value can be used for the resource allocation problem to enhance the efficiency of SDVN.
4. We abstract underlying resources dynamically and the abstracting problem is associated with the trust value of vehicles, i.e., high credibility vehicles deserve more privileges in resource allocation process. We use multi-commodity flow algorithm in SDVN to allocate resources efficiently and achieve better load balance and higher capacity with the guarantee of traffic safety.

The rest of this paper is organized as follows. Section 2 lists some related works on SDVN and trust management architecture respectively. Section 3 shows the three-tiered architecture of trust management for SDVN and illustrates the functions of each layer. The basic methodology in this architecture is introduced in Section 4. Section 5 illuminates specific procedures which reveal how this redesigned architecture works. The security analysis is discussed in Section 6. In Section 7, SDN technology is adopted to serve those high credibility vehicles and propose a resource abstraction algorithm to enhance performance. Numerical results are shown in Section 8. Section 9 clarifies the conclusion and future work.

2 Related works

A few pioneering explorations have discussed the feasibility of SDVN. Literature [3] illustrated the heterogeneous vehicular communication by using SDN and provided a framework in heterogeneous vehicular communication. In [5], the author worked on the transmission delay cost reducing problem during the download of cellular networks on the control plane. The related works in SDVN so far have

been summarized in [6], meanwhile, it has pointed out the key requirements and challenges of SDVN.

The main idea of physical resources allocation problem is to divide physical network into many mutually isolated virtual networks through virtualization technology, which can share the same underlying physical resources [10], which is the scope of control plane optimization. The virtualization problem can be converted to abstracting physical resources into virtual nodes and links, namely, we should set up virtual networks in which could provide different kinds of services and maximize resource utilization. Due to the diversity of virtual networks, the mapping algorithm is the NP-hard problem. Based on the idea of bandwidth integration and traffic engineering, the authors in [11] used path migration to map nodes and links separately. Considering the aim of the multi-commodity flow problem is to minimize costs in multi-source and multi-destination condition, therefore we can apply this algorithm to design some mathematical modeling for mapping problem.

Although several studies have researched on SDVN, whereas none of them considers a case that malicious vehicle will put forward fake resources requirements to deteriorate network performance, and no literature has been described the association between the trust management and SDVN before. Hence, in this paper, we combine the SDVN with the trust management for the first time.

The trust management for vehicular networks can be categorized into centralized, decentralized, and blockchain-based decentralized trust management. In the centralized trust management systems, several literatures have studied the architecture and operation process. A fully trusted central server with powerful processing capacity is deployed to acquire, calculate and store the trust value of all vehicles in [7, 12]. A reputation-based announcement scheme was proposed in [13], in which vehicles perceive traffic relevant events and announce them to neighbors. After estimating the message credibility sent from neighbors, all the vehicles will feedback the credibility report about their neighbors to the central sever. The central server will update reputation values of vehicles in the light of the feedback reports. Whereas, all the schemes above are supposed to deploy a fully trusted central server which cannot be compromised by attackers, and the single point of failure is still a fatal problem to this architecture. Meanwhile, with the increasing of the number of intelligent vehicles, the central server cannot always satisfy the rigorous QoS demands.

In the decentralized trust management system, a data-centric trust management scheme was proposed in [14], in which each receiver will estimate each piece of the received data, and aggregate them to judge the traffic events. This kind of scheme was executed on the vehicle side, whereas the malfunctions might appear due to the limited observation conditions of vehicles. In [9], each RSU was employed for trust management, the RSU collected the rating uploaded by the vehicles and used a specific algorithm to calculate trust value of each vehicle. Whereas the storage information in RSUs might be incomplete and inconsistent which occurs to that the different RSUs may calculate different trust value of the same vehicle.

Refer to the blockchain-based decentralized trust management systems, literature [15] employed a joint proof-of-work (PoW) and proof-of-stake (PoS) consensus mechanism to reach consensus among RSUs in trust management for vehicular networks. Although the scheme is novelty, it lacks practicality, and largely wastes the computation resources and reduces the throughput. Since this scheme was based on the public chain network, the user privacy cannot be guaranteed. Different from the schemes above, in this paper, we design a trust management scheme for SDVN and propose a new consensus algorithm to speed confirmation time. Based on the trust value derived from the trust management system, RSU can allocate more physical resources to those vehicles who have higher credibility.

3 System model

There are three layers in our proposed decentralized trust management system model, which are rating generation layer, blockchain infrastructure and cloud area as illustrated in Fig. 1. The respective function that each layer undertakes will be elaborated on in the followings.

Cloud area The main entity in cloud area is a trustworthy CA (Certificate Authority). CA can cooperate with vehicle administrative office to access vehicle relative information. CA undertakes three functions as follows:

1. *Account registration:* To prevent illegal users from entering the blockchain network, each vehicle must enroll from CA to gain authority to access the network. Real identity information must be sent from vehicles to CA (e.g., plate number), then each vehicle will be assigned a unique account for the network to communicate with other vehicles or RSUs.
2. *Account audit:* Because of CA stores the real identity information of vehicles, hence CA is responsible for auditing. Once some vehicles behaviors are malicious, CA can implement some punishment schemes to warning the vehicles.
3. *Account revoke:* Once the trust value of a vehicle is lower than a threshold, CA will revoke the vehicle account and announce the whole network so that all the entities running on the vehicular networks will not provide services for the forbidden account. Trust value

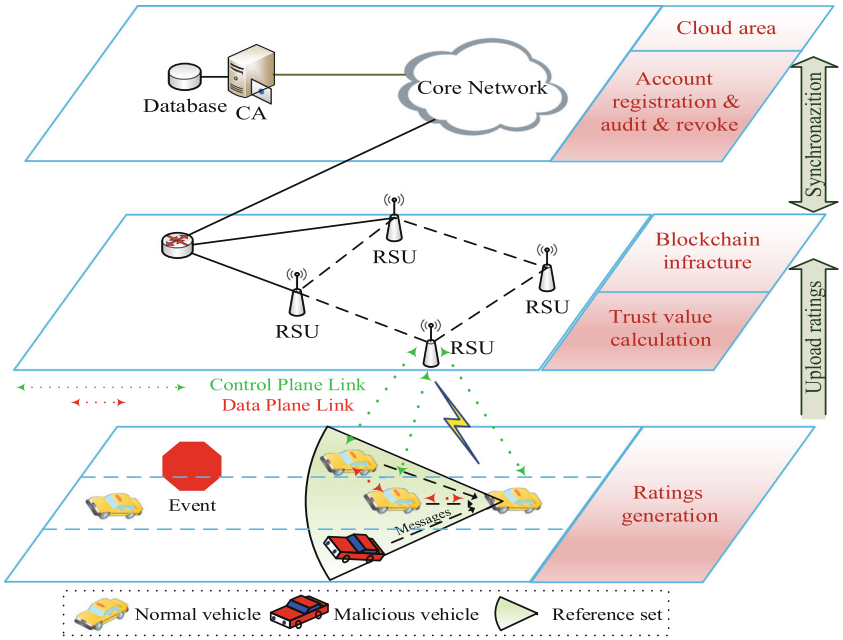

Fig. 1 System model

is generated based on the rating that recorded in the distributed ledger, and is also constantly renewing.

In addition, CA can use traditional relationship database (i.e., Oracle) to store account-relevant information.

Blockchain infrastructure Blockchain infrastructure comprises of RSUs running blockchain clients and participating in consensus process. Each vehicle should upload the rating to the nearest RSU periodically. RSUs mainly undertake four aspects in this layer

1. *Rating collection:* Each RSU has the responsibility to collect rating and message credibility from vehicles in its jurisdiction. The rating and message credibility are produced and calculated by vehicle according to the received road-relevant messages from neighbors.
2. *Ledger update:* RSUs collect rating and messages credibility from vehicles within its jurisdiction in a certain time and batch them into a block. All RSUs should add the corresponding blocks into the distributed ledger and reach consensus through a joint PoS-mPBFT consensus algorithm.
3. *Trust value calculation:* All RSUs are responsible for calculating trust value of vehicles based on the distributed ledger. Because of the data stored in each RSU's ledger are the same, hence the calculated trust value of a specific vehicle in any RSU is the same. All RSUs reach consensus about the trust value of vehicles. All the registered vehicles can inquiry trust value of other vehicles from any RSU.
4. *Dynamic underlying resources allocation* A RSU acts as a controller in SDVN, and is responsible for mapping physical network into virtual networks according to the underlying requirements. Once congestion occurs, RSU should respond to the congestion.

Rating generation layer The main entities in rating generation layer are intelligent vehicles. These vehicles are equipped with on-board sensors, computers and communication devices for data gathering, processing and uploading. Traffic-relevant events can be automatically detected and transmitted among vehicles relying on the on-board devices. Long Term Evolution Vehicle-to-Vehicle (LTE-V2V) or Dedicated Short-Range Communications (DSRC) can be employed in communication process. However, with the ever increasing of intelligent vehicles, it has been more difficult to have a fine access management among these different communication technologies. The scheme in [16] can be adopted to optimize the control and management of the network infrastructure. The receive vehicle only receive messages from reference set, which is a set of high relevance traffic safety vehicles that report traffic-relevant messages ahead of the receiver and not exceed the traffic event location, as Fig. 1 illustrated. If a vehicle passes the event location, the messages it reports are meaningless and might mislead the road condition judgment of the receive vehicle.

Each receiver can receive different kinds of messages reporting on the same event, however, not all messages are credible because of the fake messages spreading from malicious vehicles or honest vehicles have malfunctions. Therefore, each vehicle should aggregate all messages from reference set about a specific event and recognize the credible one. Discriminative model can be adopted for

recognition, e.g., the majority rule. Thereafter, vehicles can upload rating to RSUs that based on the message credibility. In addition, some high credibility vehicles can acquire higher throughput through SDN technology.

4 Methodology

As aforementioned in the previous section, blockchain technology is deployed in the blockchain infrastructure layer. Indeed, blockchain works as a promising technology can realize decentralized systems and enhance network security, and it is essentially a distributed ledger. The working mechanism of blockchain is illustrated in Fig. 2. The key novelty of blockchain is its ability to enable sharing of a common ledger amongst trustless (and anonymous in some cases) parties using distributed consensus protocols and cryptographic puzzles. Transactions or other records are stored in the so-called blocks, and different blocks are chained by hash pointers which provide tamper-proof, traceability and other security properties. However, there are many types of blockchain-based networks, and generally can be divided into three categories, i.e., public chain network, private chain network and consortium chain network. So we must adopt an appropriate type of blockchain to deploy it in the SDVN. In the public chain network, all users can participate in the network without registration and authentication, access the distributed ledger and try to mine a block into the ledger. Because the vehicles and RSUs have ever registered from DMV (i.e., vehicle administration office), the public chain network is not suitable for VANET trust management. In public chain network, a block is invalid until another six blocks are generated based on the mined block. Therefore, the timeliness of the trust update cannot be satisfied. Users in the private chain network are all registered and permissions of them are strongly restricted, i.e., the read and write permission. Therefore, in the private chain network, we assume all the participants are legitimated and security, and only consider the fault caused by communication process (e.g., packets cannot reach the destination). Apparently, it is unreasonable to adopt this chain network in VANET trust management, for we cannot reckon all the vehicle and RSUs are secure and only transmission fault occur. Besides, different brands of vehicles have different manufacturing structures, the potential security risks cannot be neglected. Hence, the consortium chain network could be adopted into the trust management, which considers both the communication and Byzantium errors.

5 Main procedures

In our decentralized trust management system model, we assume CA is highly reliable which means it will not be compromised by attackers. RSUs are relatively credible which means it might be attacked by some attackers, whereas the quantity of the attacked RSUs is minority. There are some malicious vehicles will disseminate reports about road conditions to other vehicles. The specific procedures will be explained as following subsections.

5.1 Account registration

Before entering the blockchain network, each vehicle must register in CA to gain authority to access the network. CA will verify the validity of each registered vehicle and assign a unique pseudonym. The pseudonym is an account that is used to communicate with other vehicles or RSUs on the blockchain network, and is generated according to the real identity information and public key each vehicle holds. Only the authorized or registered vehicles can upload data, query information stored in the distributed ledger on the blockchain network and query trust value of vehicles from other RSUs. Firstly, a vehicle uses asymmetric encryption algorithms (e.g., RSA, Paillier etc) to generate a key pair (Pk_usr, Sk_usr) in which the Sk_usr is held secretly by vehicle itself, whereas the Pk_usr and the license plate number will be encrypted by PK_CA and then sent to CA. For the sake of uniqueness of each pseudonym, CA uses

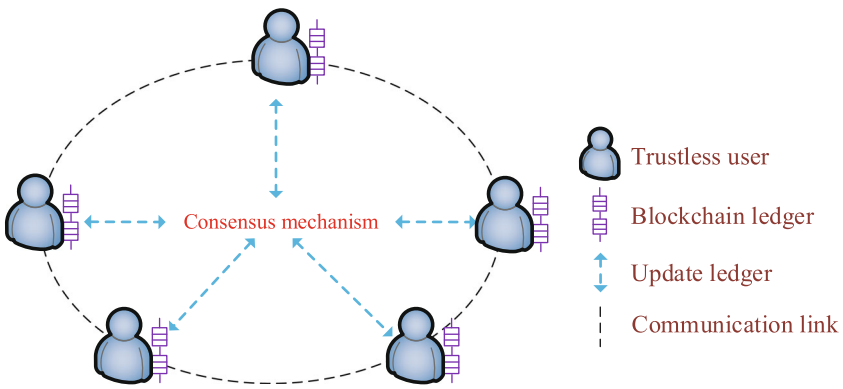

Fig. 2 Working mechanism of blockchain

salt hash algorithm to generate an account as Eq. 1 shows. The σ is a big random number generated by CA to defend dictionary attacks and brute force attacks for the protection of user privacy.

$$Account = hash(Pk\_usr + plate\_num + \sigma). \quad (1)$$

5.2 Rating calculation

We classify the vehicles into three categories according to roles, which are privilege vehicles, enterprise vehicles, and individual vehicles respectively. The privileges vehicles, denoted as C_1, are some vehicles that designed and manufactured for specific scenarios like police car and public transport. This kind of vehicles has a stable driving trajectory, equipped with stronger communication and security equipment and can defend more risks than normal vehicles. The enterprise vehicles, denoted as C_2, are those vehicles that served for companies, and are always been filed and oversaw by the company, e.g., taxi and logistics companies. This kind of vehicles always has higher credibility than individual vehicles but lower than those privilege vehicles. The individual vehicles, denoted as C_3, are those vehicles that designed and manufactured for families. This kind of vehicles are always lack of regulatory mechanism, hence easily attacked and hacked. Once been hacked, the car owners are difficult to perceive, therefore this kind of vehicles has the lowest credibility compared with the former two kinds. Based on the above, we can define a constant confidence value for these three kinds of vehicles as Eq. 2 shows. The confidence value represents the initial credibility of this kind of vehicles holds, which can be employed to calculate message credibility in the later.

$$OC_p = \begin{cases} \gamma_1, & p \in C_1 \\ \gamma_2, & p \in C_2 \\ \gamma_3, & p \in C_3 \end{cases} \quad (2)$$

A vehicle might receive diverse traffic-relevant messages from different vehicles in a certain time interval. We assume there are M kinds of road-relevant events in total, and the number of vehicles in reference set is J. A receiver will separate all receiving messages from the reference set into M groups, i.e., $\{g_1, g_2, ..., g_M\}$. Each group is a set of vehicles which have reported a specific message e_i (e.g., there is a traffic jam at road segment A or there is no traffic jam at road segment A). Whereas not all vehicles in the same group have the equal message credibility, the receiving time and the distance from reporter to event position will significantly influence the message credibility. Therefore, we can define the credibility of a specific message as Eq. 4 shows. Each message credibility is associated with a tuple W_k^j in Eq. 3.

$$W_k^j = (d_k^j, t_k^j, OC_p), p \in (C_1, C_2, C_3). \quad (3)$$

where d_k^j is the distance between vehicle v_j and the event location reported in message k. The t_k^j is the message received time from the receiver side, and OC_p is the initial confidence value that vehicle j holds. We use subject logic to calculate message credibility in this paper [17]. Therefore, we define the message credibility is c_k^j in Eq. 4, and which represents the message k in group g_k reported by vehicle v_j, and is the weighted sum of distance and timeliness relevance and initial confidence parameters. The m_k^j is the timeliness of message k from vehicle j which is referred in [9]. The η is the preset balance coefficient between distance and timeliness, α, β and ε are the preset parameters to control the upper bound and lower bound. The d_k^j is the distance between vehicle v_j and the event location reported in message k. m_k^j is the timeliness of message k from vehicle j which is referred in [9]. If the receiver firstly receives the message k from a vehicle v_n, t_f is set to be the received time, and once received the same message from vehicle j, t_k^j will be set to the received time. By the way, if a vehicle j doesnt report event k, c_k^j will be set to zero.

$$c_k^j = \eta e^{\alpha d_k^j} + (1 - \eta) m_k^j + OC_p. \quad (4)$$

Accordingly, we can obtain a message credibility set $C = \{C_1, C_2, .., C_M\}$ and each element is a credit vector comprising of message credibility. For example, vector $C1$ contains $\{c_1^1, c_2^1, ..., c_j^1\}$, which represents the messages credibility for each vehicle for event e_1.

Because of attackers are unawareness of the real road conditions and disseminate fake road-relevant messages, therefore the message credibility based on the specific event that an attacker report will be largely different from those come from trustworthy vehicles, and different from malicious vehicles. Meanwhile, perception ability and scope are similar in on-board devices, therefore the reported distance and time d_k^j, t_k^j will be similar among honest vehicles. In addition, nevertheless attackers would disseminate fake messages, the quantity is still minority. We assume the ratio of malicious vehicles in reference set will not exceed half. Based on the three assumptions above, the receiver can calculate the event rating.

We define a preset threshold Thr, if a certain percentage reported-vehicles do not exceed the threshold, we can reckon this event is fake. If exceed, we then calculate the floating message credibility range of a specific event according to Eq. 6. If there exists a message credibility exceed or below this floating range, we can reckon the reported vehicle is malicious. If the number of vehicles in the floating range is majority compared with the number of reported vehicles, then, we can recognize the message k is true. In a period of time, a receiver might receive different types of messages from different vehicles. We define the

rating of a specific vehicle is the percentage of correct events reported by this vehicle accounts for the proportion of the number of all reported incidents.

$$f = \sum_{i=1}^{n} \frac{c_k^j}{n}, \tag{5}$$

$$Fr = [(1-C)f, (1+C)f]. \tag{6}$$

After calculating the rating of each event, the receiver would upload calculation results to RSU. Due to the limited space of this paper, we are no longer describe specific format in detail here.

5.3 Consensus mechanism

Traditional Bitcoin networks depend on miners to collect transactions generated from users, batch them into a block, and then try different nonce to solve a cryptographic problem, i.e., find a specific hash value which is relevant to the previous block. After solving the same cryptographic problem, miners will add the current block into the distributed ledger. The above method is called PoW consensus mechanism [18], and this mechanism sacrifices computation resources to gain fairness among miners. Whereas, huge computation cost, easily-forked and low throughput are still remaining disadvantages based on this consensus mechanism. PoS [15] is another consensus method, all transactions will be sent to a stakeholder which is chosen according to assets. The basic idea of PoS is the one with majority assets are willing the system to operate normally, once the system is attacked, its own assets will be damaged seriously than those poverty users.

The above two consensus mechanisms are all designed for the public blockchain network, in which the validity and credibility of participants cannot be measured and audited. Whereas in a private or consortium blockchain network (i.e., all or partial users are registered and can be audited), we can use other consensus algorithms to reach consensus, e.g., PBFT (practical Byzantium fault tolerance). In PBFT [19], a leader is chosen to lead the current clients to reach consensus on collected events or transactions. In most scenarios, the leader sequence is calculated based on the total quantity of current valid clients through modular arithmetic, which is more easily to be exposed to the attackers. Considering the RSUs are along the road, easy to expose to attackers, hence a joint PoS and mPBFT (modified PBFT) consensus scheme is proposed in our system.

Generally speaking, RSUs are usually more credible, stable and with more powerful calculation ability than intelligent vehicles to process and store data, therefore RSUs are served as endorses in blockchain infrastructure layer. The stakes that each RSU holds are estimated according to the sum of rating in real events.

The consensus procedures can be divided into three phases as depicted in Fig. 3:

5.3.1 Leader selection

Leader selection phase can be divided into two procedures as follows:

Stakes collection The stakes RSU_i holds in a certain time can be represented as Eq. 7, which is the sum of percentage of real events each vehicle reported. And we set an upper bound F_{max} to avoid compromised attackers deliberately boost its stakes.

$$F_i = min\left(\sum r_k, F_{max}\right). \tag{7}$$

Periodically, each RSU should broadcast the calculation result to all other RSUs and forward other newly arrived

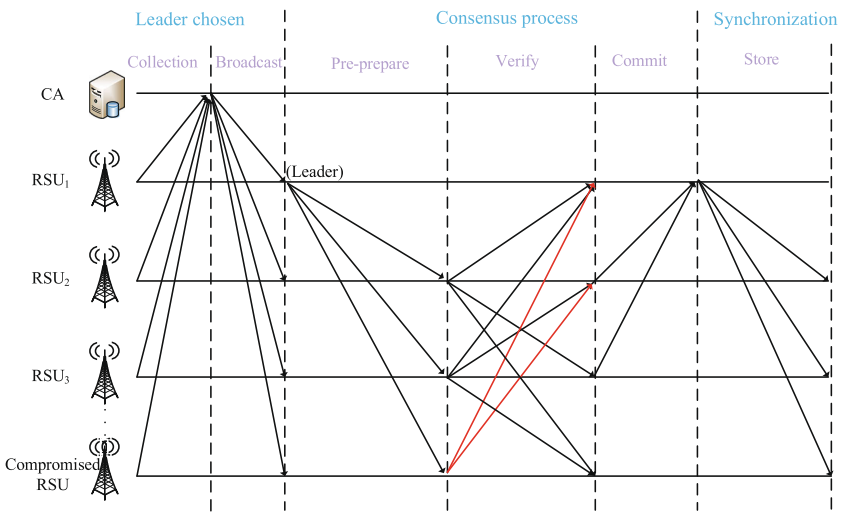

Fig. 3 Consensus procedures

calculation results of other RSUs in the fully decentralized network, whereas this behavior will no doubt seriously enlarge the communication overheads. Meanwhile, the consensus about which RSU holds the most stakes also takes a long time. Actually, the solution to the aforementioned problem has been proved costly in [8].

View from the existing problems above, we can use the fully trustworthy CA to collect, compare and choose the leader (i.e., RSU with the most stakes). Each RSU calculates stakes based on its own collected information and sends it with calculation result to CA.

Broadcast CA will compare the calculation results from each RSU and select the one with the most stakes. Once some RSUs are with the same stakes (e.g., reach the upper bound), the leader will be chosen randomly among them. When the comparison process is completed, CA will send all the collected information to the leader, and inform other RSUs about the current leader.

5.3.2 Consensus process

The consensus process is carried out by followers (i.e., authorized RSUs) and the leader with the most stakes chosen by the CA. The leader encapsulates all rating of vehicles into a block with a time order, and broadcasts to other followers. In PBFT, the fault tolerance rate is $\frac{1}{3}$, which not only consider the transmission error (packets cannot reach destinations timely) but also the malicious nodes that can fake the exchanged information. The specific details are elaborated described in [19]. Whereas in the vehicular networks, all RSUs are working in the private network, i.e., under the stable circumstance, meanwhile, transmitted by the wired network. Hence, we assume all the packets that sent from senders could reach the corresponding destinations. Based on this assumption, we could simplify the consensus process. At the beginning of the verify phase, those f attackers can choose to modify the block content, and then broadcast to other RSUs. For the ease of mutual verification and supervision, follower will audit the block data and broadcast the verification results with signature to other followers. Each verification result comprises of (audit result, verification result, signature, reply). Followers audit whether their own data has been tampered and verify whether illegal vehicles are included in this block and reply to other followers with their signatures. Once all followers receive the unmistakable audit messages from others, they will send commit messages to the leader. At the end of this phase, once a RSU receive $f + 1$ replies then it will enter the commit phase. The modified consensus algorithm can provide at most $\frac{n-1}{2}$ out of a total of n RSUs are simultaneously faulty.

5.3.3 Synchronization

The synchronization phase is to guarantee the leader is in normal operation. Once the leader has been compromised, a so-called view change process can be employed, which select another leader through CA with the help of feedbacks provided by RSUs. After that, the leader will write the block into the blockchain network and send to all other authorized RSUs for synchronization and storage.

5.4 Trust value prediction

Trust value of vehicle is generated based on the history rating that have been written in the blockchain network, and is updated when new rating are introduced in. Most literatures have been concentrated on how to design appropriate methods to reflect real trust value of vehicles. Nevertheless, none of them have considered how to predict the trust value changes. In the view of the trust value generation process, trust value cannot have a suddenly big change, the trust value of a vehicle are slowly up or down, which provides the theoretical fundamentals to predict the trust value of vehicles. We intercept past several trust values calculated from the distributed ledger as sample data and use an array M to denote it as

$$M = \begin{bmatrix} r_j^i & r_j^{i+1} & \dots & r_j^{i+n} & 1 \\ r_j^{i+1} & r_j^{i+2} & \dots & r_j^{i+n+1} & 1 \\ \dots & \dots & \dots & \dots & \dots \\ r_j^{i+n} & r_j^{i+n+1} & \dots & r_j^{i+2n} & 1 \end{bmatrix} = \begin{bmatrix} M_i & M_{i+1} & \dots & M_{i+n} \end{bmatrix}^{\mathrm{T}}, \quad (8)$$

where j denotes the j-th vehicle and 1 is a constant. Each element in M except 1 represents the trust value of the specific vehicle j.

The prediction trust value of vehicle j can be calculated according to

$$C_j^{s+1} = \sum_{i=1}^{n} \alpha_i \cdot r_j^i + \beta. \quad (9)$$

We set vector A is regression coefficients array $[\alpha_1, \alpha_2, ..., \alpha_n, \beta]$ and its elements can be learned through the least squares method.

We use the latest trust value to fit past several trust values of a specific vehicle to calculate regression coefficients array A. We set $Y = [r_j^{i+n+1}, r_j^{i+n+2}, , r_j^{i+2n+1}]$ are the real latest trust values of a specific vehicle, therefore, correspondingly the fitted trust value are $[M_x \cdot A^{\mathrm{T}}, M_{x+1} \cdot A^{\mathrm{T}}, , M_{x+t} \cdot A^{\mathrm{T}}]$ according to Eq. 9. We define E_A as follows

$$E_A = (Y - M \cdot A^T)^{\mathrm{T}} \cdot (Y - M \cdot A^T). \quad (10)$$

The regression coefficients array A can be obtained when E_A achieves the minimum through (11).

$$\frac{\partial E_A}{\partial A} = \frac{(Y - M \cdot A^T)^{\mathrm{T}} \cdot (Y - M \cdot A^T)}{\partial A} = 0. \tag{11}$$

6 Security analysis

As aforementioned, in our proposed architecture, we consider two kinds of attack patterns, i.e., compromised RSUs and malicious vehicles.

RSUs are deployed alongside the road which will expose to attackers. We set there are no more than half of RSUs are fault, of which will tamper block content, deny service, and disseminate fake messages. Owing to the block structure and self-check function, the tampered block content could be identified. Once RSUs disseminate fake messages or deny service, the mPBFT mechanism will be adopted to solve these situations.

Malicious vehicles will report false road conditions to neighbors which could be resulting in traffic accidents. The weighted sum of time and distance are taken into consideration to evaluate message credibility, and floating range to identify message accuracy.

7 Physical resources abstracting

Through the trust value obtained from the proposed trust management system, each RSU holds the same trust value for every enrolled vehicle. As aforementioned in the introduction, malicious vehicles can put forward fake resources requirements which will disrupt the normal operation of SDVN. Through trust management system, we can obtain the trust value of each vehicle, and assign more resources to those high credibility vehicles, which guarantee the network security and efficiency. Therefore, in this section, we focus on enhancing the load balance and efficiency associated with vehicle credibility in SDVN on the control plane.

7.1 Problem definition

For the ease of expression, we consider a scenario that multiple vehicles served by one RSU, in which RSU acts as a controller to manipulate several vehicles requirements. RSU has enough processing capacity to process several service requirements from vehicles. Therefore, the problem is equal to allocating resources to vehicles effectively.

We can transform the optimal resource allocation problem into a linear programming problem. For ease of description, we put the underlying physical network as an undirected graph $G=(V,P)$.

The set of served vehicles is denoted by V, and the set of physical paths between two vehicles is represented by P. Based on the physical network, the virtual subnet $G' = (V', P')$ can be constructed. Due to the different vehicles have different requirements [2], therefore it is reasonable to assume that there are various types of services in a virtual network. We divide the multiple services into K classes which means there are K data streams in a virtual subnet.

The difficulty in mapping problem is to meet the requirements Rq in each link. Namely, we need to design a reasonable mapping algorithm to improve the capacity of the network with limited physical resources. From the definition and analysis above, we mainly consider the vehicles, link, bandwidth and trust value of vehicles to denote network application requirements as Eq. 12 shows.

$$Rq = \{(s_i, t_i, d_i^{\rho_i}), 1 \le i \le k\}, \tag{12}$$

where s_i and t_i is the source vehicle and destination vehicle of class K requirements respectively, d_i is the bandwidth requirement between source vehicle and destination vehicle, ρ_i is the trust value of the destination vehicle.

7.1.1 Single-path mapping

In network computing, we deal with various network data streams as separated commodity streams. For the established virtual subnet, we denote it as $G'(V', P')$, where V' is the set of virtual network nodes mapped by physical vehicle V and also a subset of V. P' is mapped by physical paths P.

Suppose that there are L physical paths in total, and the maximum number of virtual links mapped by a physical path is K. A physical path p_i can support one or more virtual links e_i , and e_i is the element of set P'. An illustration of single-path mapping is shown in Fig. 4. We can represent p_i

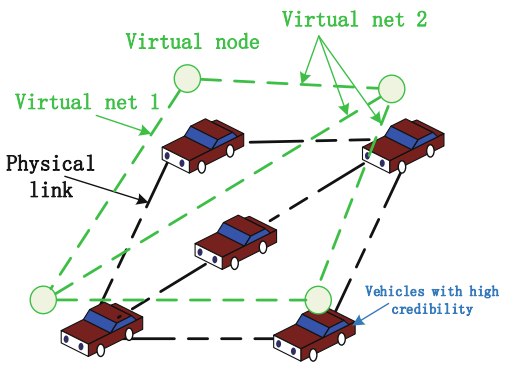

Fig. 4 Illustration of single-path mapping

and e_i as $\{p_i|pi \in P, 1 \leq i \leq L\}$ and $\{e_i|e_i \in P', 1 \leq i \leq K\}$. The optimization problem is defined as:

$$\min \quad \max_{e_i \in P'} \{w(e_i)\} \tag{13a}$$

$$\text{s.t.} \sum_{1 \leq k \leq K} u(e_k) \leq x(p_j), 1 \leq j \leq L, \tag{13b}$$

$$NU^l = B^k, l \in [1, L], k \in [1, K], \tag{13c}$$

$$x(p_i) \geq 0, \forall p_i \in P. \tag{13d}$$

where we have

$$\sum_{i=1}^{K} w(e_i) = \sum_{i=1}^{K} \frac{u(e_i)}{x(e_i)} = \frac{\sum_{i=1}^{L} u(p_i)}{\sum_{i=1}^{L} x(e_i)}. \tag{14}$$

N is a K-Rows and L-columns associative matrix of G', the element value of N is taken 0 or 1. N_{ij} is 0 represents e_i and p_j have no mapping relationship, vice-versa.

Equation 13a minimizes maximum load intensity. In Eq. 14, $w(e_i)$ is the load intensity of the link, which is equal to the ratio of the occupied bandwidth of the link e_i. $u(e_i)$ is the bandwidth occupied by the link e_i, $x(e_i)$ is the bandwidth capacity of the link e_i, $u(p_i)$ is the occupied bandwidth of the physical path p_i. $x(p_i)$ is the bandwidth capacity of the physical path p_i. A physical path p_i may be mapped to multiple virtual links e_1, e_2, e_i. They occupy the bandwidth capacity are $u(e_1), u(e_2), u(e_i)$. Equation 13b denotes the sum of the bandwidth occupied by all virtual links on path P that are not greater than the bandwidth capacity of path P, which is the physical path bandwidth capacity constraint, and ensures the physical link will not be overloaded. Equation 13c ensures that the network can carry all K kinds of streams to meet the various corresponding bandwidth needs. U is the size of $L * 1$ actual occupied physical bandwidth vector, B is the size of $K * 1$ network application bandwidth demand vector.

The above algorithm is based on the assumption that the physical link has not been occupied by other links before virtual links, hence network association matrix is known. In the actual virtual link construction process, the available bandwidth on physical link is usually not equal to its bandwidth capacity because of some interference. Therefore, we propose a multiple-path mapping algorithm for resource allocation.

7.1.2 Multiple-path mapping

In multiple-path mapping, a physical link can support one or more virtual links, meanwhile a virtual link can be mapped to multiple physical links [20]. The illustration of multiple path mapping is shown in Fig 5.

Considering the real-time dynamic changes in SDVN, we define $w(e_i)$ is the link load intensity, which equals to the ratio of occupied bandwidth and the capacity of link e_i. The objective function is to minimize the maximum link utilization under meeting the bandwidth requirements of network applications. Hence, the multiple-path mapping problem is defined as:

$$\min \quad \max_{e_i \in P'} \{w(e_i)\} \tag{15a}$$

$$\text{s.t.} \; u(e_k) \leq \sum_{i=1}^{L} x(p_i), 1 \leq k \leq K, \tag{15b}$$

$$\sum_{i=1}^{K} \sum_{j=1}^{L} x(p_{ij}) \leq x(e), e \in P', \tag{15c}$$

$$0 \leq x(p), p \in P. \tag{15d}$$

where we have

$$\sum_{i=1}^{K} w(e_i) = \sum_{i=1}^{K} \frac{u(e_i)}{x(e_i)} = \frac{\sum_{i=1}^{K} \sum_{j=1}^{L} u(p_{ij})}{\sum_{i=1}^{K} x(e_i)}. \tag{15e}$$

$x(p_{ij})$ represents the bandwidth of the link e_i mapped to the physical path p_j.

7.2 Iterative solution

Heuristic algorithm is often used to solve the NP-hard problem. Our proposed heuristic algorithm has two stages. First, according to the single-path mapping method, we could obtain an initial feasible solution. Then, based on the initial feasible solution, we use the dichotomy to solve the problem through multiple iterations.

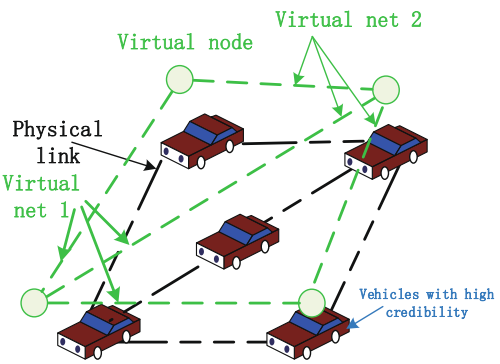

Fig. 5 Illustration of multiple-path mapping

Before accessing initial feasible solution, we need to give some definitions:

Definition 1: Available bandwidth (p) of link p is

$$c(p) = \min_{e \in p} c(e). \tag{17}$$

$$c(e) = x(e) - u(e). \tag{18}$$

Definition 2: Fragility $\alpha(p)$ of link p:

$$\alpha(p) = \max_{e \in p} \alpha(e). \tag{19}$$

where $\alpha(e)$ is the number of virtual links mapped into different physical links. For ease of expression, we use e_{ij}^k represents the virtual link between source node i and destination node j. C_{ij}^V denotes the bandwidth requirements. p^k is the set of e_{ij}^k mapped into physical links.

First, we sort C_{ij}^V by non-increasing subset. Then we use single-path mapping method to choose one physical link p for mapping e_{ij}^k. Meanwhile this physical link p needs to meet:

$$C_{ij}^V \le c(p). \tag{20}$$

$$\alpha(p) = \min_{p \in P^k} \alpha(p). \tag{21}$$

Next if p cannot meet $C_{ij}^V \le c(p)$, which means that the network can not provide enough physical links for virtual link resources, the QoS of APs may deteriorate, which means the system could not provide satisfied service. Considering that, we use multiple-path mapping method to allocate bandwidth resources into virtual links according to the value of $\alpha(p)$ in set P^k.

In the initial algorithm, we first use single-mapping method to allocate physical resources. But once the system does not have enough physical resources, we use multiple-mapping method. Initial algorithm can get an initial feasible solution but only consider the local balance, thus if we want to consider from the entire network, we should optimize the initial feasible solution using dichotomy.

Optimization processes are shown as follows:

At the beginning, we set the initial β=1;

Step 1: Assuming there is a virtual link, which maximum load balancing value is $w(e^*) = \max_{e \in E}\{w(e)\}$;

Step 2: For a virtual link e^*, define its path set as P^*, it exits $P^* = \{p : e^* \in p, p \in P\}$. We take $\{p^* \in P^*, p^* \in P1, p' \in P^k\}$ to meet the condition $c(p') = \max_{p \in P^k} c(p)$;

Step 3: Set parameters $\overline{\Delta} = 0$, $\underline{\Delta} = c(p')$ and $\Delta := \frac{\overline{\Delta}+\underline{\Delta}}{2}$;

Step 4: $\forall e \in p^*, u(e) := u(e) - \Delta$; $\forall e \in p', u(e) := u(e) + \Delta$, calculate the maximum link load value e, which is compared with the maximum load value of e^* in step 1, if $w(e^*) \le \max_{e \in P'}\{w(e)\}$, then execute the next step, else jump to step 6;

Step 5: Calculate $u(e) := u(e) + \Delta, \forall e \in p^*$ and $u(e) := u(e) - \Delta, \forall e \in p'$ and set $\Delta := \frac{\overline{\Delta}+\Delta}{2}$, then go back to step 4.

Step 6: If $w(e^*) - \max_{e \in P'}\{w(e)\} \le \beta$, then $\beta = (w(e^*) - \max_{e \in P'}\{w(e)\})/2$, go back to step 1. Otherwise the algorithm achieves the convergence and gets the optimal solution when iteration is over.

8 Evaluation analysis

8.1 Evaluation on the proposed architecture

We deploy our scheme on Spyder based on python 3.6. Specific parameters are shown in Table 1, which refer to [9, 15] (Fig. 6).

We set ten kinds of traffic events in our system, three kinds of events are fake and seven are real. Malicious vehicles will disseminate fake messages on the road, by contrast, honest vehicles will transmit road-relevant messages as they observed. The trust value of vehicles is generated from history rating recorded in the distributed ledger. As illustrated in Fig 7, the prediction accuracy is continuously improved with the increasing of the number of record numbers, and can achieve 87.9% when the number of training data is 35. With the improvement of prediction accuracy, vehicles can choose to trust or doubt other vehicles in advance.

The consensus confirmation time of the joint PoS-mPBFT is little less than PBFT as illustrated in Fig 8. Because of the PoS-mPBFT simplify the prepare and commit phase compared with [19], hence we can shorten confirmation time about 10% compared with PBFT. The transactions collecting time is not counted in here, for the collecting phase is the same as [19].

Table 1 Key parameters

Parameters	Values
Vehicle Number	25
Vehicle number in reference set	between 0 and 10
Distance between vehicles	Uniform distribution
Number of reference Set	10
Message Group=10	10
Number of malicious vehilcles	3
η	0.5
C	0.2

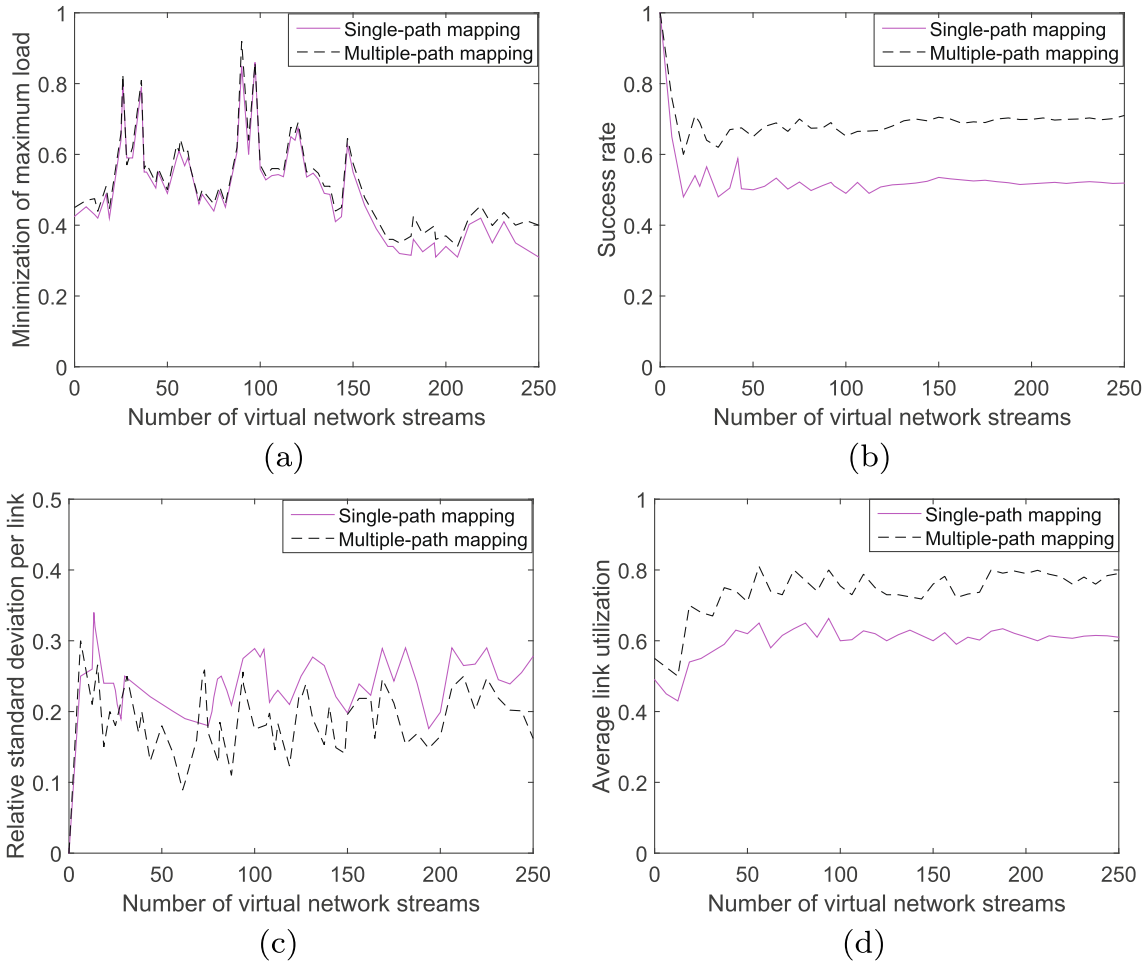

Fig. 6 Comparison of single-path mapping and multiple-path mapping

8.2 Evaluation on MPM vs SPM

The multi-commodity flow model is simulated on Matlab. Specific parameters are shown in Table 2, which refer to [21]. We compare MPM (Multiple-Path Mapping) algorithm with SPM(Single-Path Mapping) algorithm. The simulation uses 14 vehicles with 18 physical paths and the requirement of each vehicle for constructing virtual network follows Poisson distribution.

The simulation compares the performance of SPM algorithm and MPM algorithm from four aspects including minimization of maximum load, success rate of virtual

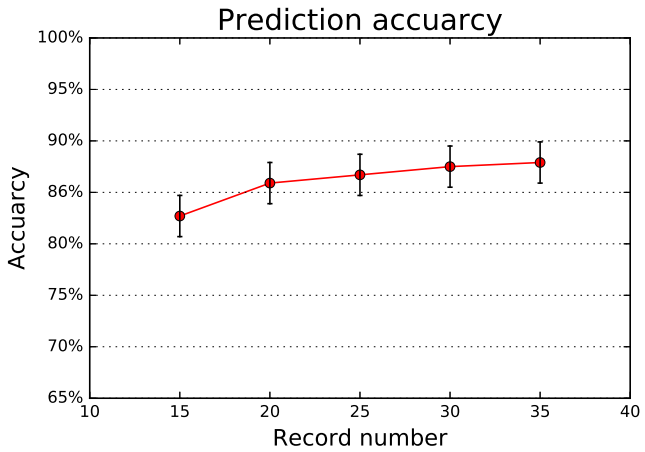

Fig. 7 Prediction accuarcy with the sample set quantity

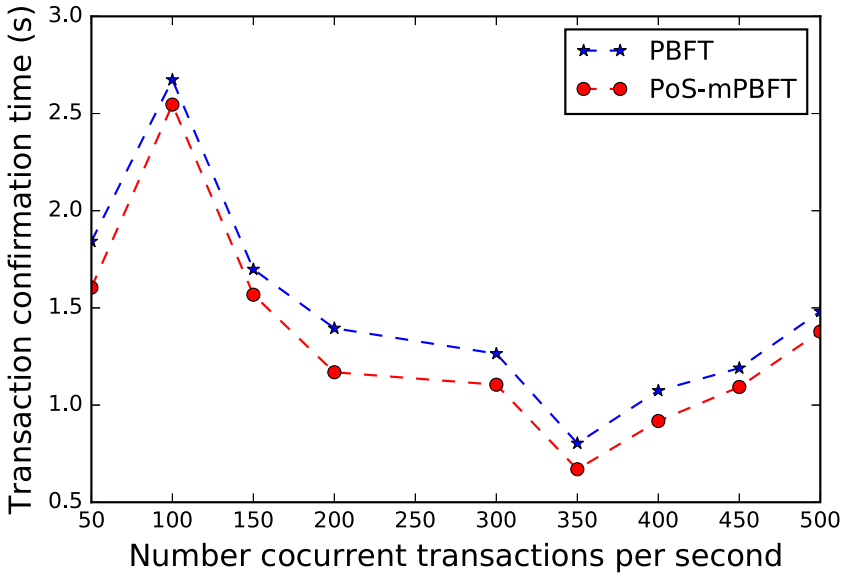

Fig. 8 Transaction confirmation on two consensus algorithms

network construction, relative standard deviation of link load and average utilization, such as Fig. 6 shows. Figure 6a shows the relationship between minimizing the maximum load and the number of virtual network applications increasing. From the illustration, we obtain that MPM algorithm is slightly higher than SPM algorithm. The minimization of maximum load achieves the peak when the number of virtual network applications between 100 to 150. This means MPM algorithm could achieve better load capacity.

Figure 6b denotes that the success rate in MPM algorithm is about 0.7 which is higher than SPM about 20%. This implies MPM algorithm could provide better safety than SPM algorithm, meanwhile, it is important in real situation for communicating effectively between APs. Figure 6c shows that relative standard deviation per link in MPM algorithm is lower than SPM algorithm about 3%, which means MPM algorithm has better load balancing when mapping physical resources to virtual network.

Simulation results are presented in Fig. 6d for average link utilization, from which it can be seen that MPM algorithm is always higher than SPM algorithm about 18%. The main reason for this is due to the fact that a virtual link in MPM algorithm can be mapped into various physical links, vice-versa.

Table 2 Simulation parameters of virtual network mapping

Parameters	Values
Number of nodes	14
Number of paths	18
Physical bandwidth	10Gbps
Virtual link bandwidth	[100mbps,1Gbps]
Virtual network application interval	30s
Virtual network average lifetime	15s

9 Conclusion and future works

In this paper, we dedicated to securing and enhancing the efficiency of SDVN. We associated the resource allocation problem with trust value of vehicles. To calculate trust value of vehicles and overcome the flaws of traditional trust management systems, we designed a consortium blockchain-based decentralized trust management system model for SDVN with three layers in considering the user privacy protection. A joint PoS-mPBFT algorithm was proposed to shorten consensus time and enhance security. At last, different from traditional schemes, prediction model was used to estimate the trust value of vehicles which is a novel prospect for future safety protection. After obtaining trust value of vehicles, we can allocate more underlying resources for high credibility vehicles on the control plane of SDVN. Meanwhile, to enhance efficiency, we converted the mapping problem in multi-commodity problem and proposed a heuristic algorithm to cope with. At the simulation section, we compared the single-path mapping algorithm with multiple-path mapping algorithm and analyzed the performances of minimization of maximum load, success rate of virtual network construction, relative standard deviation of link load and average utilization. Simulation results demonstrated that our proposed algorithm could provide better load capacity and balance in SDVN.

In the future, we may focus on security and efficiency enhancement of SDVN. Although security and efficiency can be guaranteed, some improvements can still be adopted.

Millimeter-wave technology can provide multiple gigabits per second in VANET, which shows great application potential. We may consider using millimeter-wave technology to provide multi-gigabit-per-second connectivity for vehicles, meanwhile, keep the security of the vehicles [22]. In the consensus mechanism, the communication overhead is $O(n^2)$. To decrease the communication load, regional compartmentalization methods might be suitable to solve this problem. In terms efficiency of SDN, all of our works are based on the control plane, while not consider the data plane. In the future, we may take into account the queue management algorithm on data plane to enhance network performance.

References

1. Chen C, Wang Z, Guo B (2016) The road to the chinese smart city: progress, challenges, and future directions. IT PROF 18(1):14–17
2. Nunes BA, Mendonca M, Nguyen X, Obraczka K, Turletti T (2014) A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surveys Tuts 16(3):1617–1634
3. He Z, Cao J, Liu X (2016) SDVN: Enabling rapid network innovation for heterogeneous vehicular communication. IEEE Netw 30(4):10–15
4. Ozcevik ME, Canberk B, Duong TQ (2017) End to end delay modeling of heterogeneous traffic flows in software defined 5G networks. Ad Hoc Netw 60:26–39
5. Li H, Dong M, Ota K (2016) Control plane optimization in software-defined vehicular ad hoc networks. IEEE Trans Veh Technol 65(10):7895–7904
6. Yaqoon Y, Ahmad I, Ahmed E, Gani A, Imran M, Guizani N (2017) Overcoming the key challenges to establishing vehicular communication: is SDN the answer? IEEE Commun Mag 55(7):128–134
7. Mahmoud ME, Shen X (2011) An integrated stimulation and punishment mechanism for thwarting packet dropping attack in multihop wireless networks. IEEE Trans Veh Technol 60(8):3947–3962
8. Li Z, Chi G, Tricia C (2014) On joint privacy and reputation assurance for vehicular ad hoc networks. IEEE Trans Mobile Comput 13(10):2334–2344
9. Huang X, Yu R, Kang J, Zhang Y (2017) Distributed reputation management for secure and efficient vehicular edge computing and networks. IEEE Access 5:25408–25420
10. Zheng Q, Zheng K, Zhang H, Leung VC (2016) Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning. IEEE Trans Veh Technol 65(10):7857–7867
11. Zhu Y, Ammmar M (2006) Algorithms for assigning substrate network resources to virtual network components. In: 25th IEEE international conference on computer communications (INFOCOM), pp 1–12
12. Lai C, Zhang K, Cheng N, Li H, Shen X (2017) SIRC: A secure incentive scheme for reliable cooperative downloading in highway VANETs. IEEE Trans Intell Transport Syst 18(6):1559–1574
13. Li Q, Malip A, Martin KM, Ng S, Zhang J (2012) A reputation-based announcement scheme for VANETs. IEEE Trans Veh Technol 61(9):4095–4108
14. Raya M, Papadimitratos P, Gligor VD, Hubaux J (2008) On data-centric trust establishment in ephemeral ad hoc networks. In: IEEE 27th conference on computer communications (INFOCOM), pp 1238–1246
15. Yang Z, Yang K, Lei L, Zheng K, Leung VC (2018) Blockchain-based decentralized trust management in vehicular networks. IEEE Internet of Things Journal early access
16. Secinti G, Canberk B, Duong TQ, Shu L (2017) Software defined architecture for VANET: a testbed implementation with wireless access management. IEEE Commun Mag 55(7):135–141
17. Muzio JC, Rosenerg IC (1986) Introduction—multiple-valued logic. IEEE Trans Comput C-35(2):97–98
18. Nakamoto S (2008) https://bitcoin.org/bitcoin.pdf
19. Castro M, Liskov B (1999) Practical byzantine fault tolerance. In: Third symposium on operating systems design and implementation (OSDI)
20. Gonzlez A, Barra E, Beghelli A, Leiva A (2015) A sub-graph mapping-based algorithm for virtual network allocation over flexible grid networks. In: IEEE 17th International Conference on Transparent Optical Networks (ICTON), pp 1–4
21. Jiang M, Wang B, Wu M (2011) Research on network virtualization and virtual network mapping algorithm. Chin J Electron 39(6):1315–1320
22. Martin-Vega FJ, Aguayo-Torres MC, Gomez G, Entrambasaguas JT, Duong TQ (2018) Key technologies, modeling approaches, and challenges for millimeter-wave vehicular communications. IEEE Communication Magazine 56(10):28 C 35