An Improved PBFT-based Consensus for Securing Traffic Messages in VANETs

2021 12th International Conference on Information and Communication Systems (ICICS) | 978-1-6654-3351-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICICS52457.2021.9464606

El-hacen Diallo*, Omar Dib†, Khaldoun Al Agha*
*Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190 Gif-sur-Yvette, France
{diallo, alagha}@lisn.fr
†Wenzhou-kean university, Wenzhou, China
odib@kean.edu

***Abstract*—With the increasing number of autonomous vehicles, more intelligent applications and services are needed to build an efficient transportation system. That cannot be achieved without having an efficient and secure model for recording and sharing traffic-related data. Because of its important features in terms of architecture decentralization, data immutability, transparency of actions, and communications security, the blockchain technology has recently been proposed to mitigate early VANETs (Vehicular Ad Hoc Networks) designs' security issues. In this work, we design a new consensus protocol based on Practical Byzantine Fault Tolerance (PBFT), aiming at proposing a secure traffic-related data sharing system in VANETs. The proposed consensus intelligently selects a set of Road Side Units (RSUs) to validate the traffic events emitted by vehicles and subsequently maintain the blockchain ledger to further exploit its immutable data. In this paper, we also introduce the concept of micro-transactions to reduce the size of the blockchain ledger and minimize the communications overhead between nodes. The performance of the proposed solution is assessed by simulating real-world VANETs' settings. The experimental results validate the proposed work's high performance in terms of blockchain throughput, latency, communication load, and storage cost.**

***Index Terms*—Blockchain, Vehicular networks, Consensus algorithms, PBFT, Blockchain simulation.**

I. Introduction

Due to the significant increase in the number of intelligent vehicles and the fast development of wireless communication technologies, collecting and exchanging traffic data has become easier than before. Nowadays, cars have become connected; they are supplied with smart and various sensors, On-Board Units (OBUs), and Global Positioning System (GPS) devices, easing their interaction with the surrounding environment. Through these smart devices, vehicles can collect records related to the traffic such as their current state (i.e., position, velocity, acceleration, etc.) [1], the temperature of the vehicle, and their driving style, etc. Besides smart sensors, modern cars are also supplied with cameras and radars; hence, making them even more adept at gathering environmental information such as monitoring neighbor vehicles' behaviors, traffic conditions, road obstacles, etc. Moreover, cars can now communicate the information with other cars through Vehicle to Vehicle (V2V) communications and transport infrastructure through Vehicle to Infrastructure (V2I) channels. Such connectivity also comprises the Road Side Units (RSUs) that construct the transport infrastructure's backbone. RSUs are usually placed along the Roads by the Transport Authority; they also collect traffic data through dedicated sensors, communicate the data between each other, receive traffic information from vehicles, facilitate the vehicles' connectivity, send traffic recommendations, and provide on-demand access to transport services.

By taking advantage of the collected traffic data, both the research and industry communities have recently made significant efforts to bring Intelligent Transportation Systems (ITS) to maturity [1]. Many applications have been developed to reduce traffic congestion and provide real-time and optimized itinerary services or announcements about the weather and traffic conditions. As a result, the vehicular network's overall quality is enhanced, and the drivers' safety is maximized.

Despite the significant improvements, there are still critical challenges hindering the development of efficient VANETs. Indeed, with the ubiquity of cybercrimes, VANETs have become more prone to attacks, whether on the network layer (e.g., DDoS, Sybil attacks) or the application layer (e.g., bogus information attacks) [2]. Furthermore, both vehicles and a portion RSUs are subject to attacks and could act maliciously against the system by spreading fraudulent messages or performing various attacks. Such behaviors affect the reliability of the data-sharing model between all the components in the vehicular network. Therefore it is crucial to implement robust protocols that ensure the transparency, immutability, integrity, and trustworthiness of the traffic-related records gathered from vehicles during their whole life-cycle.

We believe that the blockchain technology and its capacity to maintain a decentralized and consistent database between untrusted entities without relying on a third party could mitigate the aforementioned security limitations in the VANET network. Lately, this technology has got much interest by both academics and technology companies and has found applications in various domains [3].

Recently, blockchain was relied on to design secure and robust VANETs architectures [4]. Essentially, blockchain technology consists of a peer-to-peer network coordinated by a consensus algorithm and agree on the ledger's state. From a data structure point of view, a blockchain is a linked list of blocks, where each block stores the hash of the previous block. Blockchain technology has emerged along with revolutionary promises toward more secure and robust applications and architecture design through its exciting properties. By design, the blockchain offers decentralization of the network, the transparency of all actions, the immutability of data once written in the ledger, and its suitability for trustless environments without relying on a single central authority.

Motivated to propose a secure framework for secure traffic-related messages management, we take advantage of the blockchain promising features to securely, immutably, and transparently trace exchanged records between vehicles and Road Side Units (RSUs) without relying on a central authority. Each RSU is a blockchain node, i.e., maintains the blockchain and validates with other RSUs the correctness of traffic-related events messages sent by vehicles. We propose an improved consensus algorithm based on the well known PBFT consensus protocol to deal with the traffic messages in VANET. We enhance the proposed protocol by intelligently selecting a few RSUs to participate in the consensus phase and introduce micro-transactions to cope with the blockchain size. We assess the performance of the proposed protocol by simulating real-world settings in the context of VANETs.

The remainder of this article is organized as follows. In section II, we present the state-of-the-art related to the blockchain based applications in the VANETs' context. Next, in section III, we give a brief overview about the VANETs' system architecture. In section IV, we describe the proposed protocol that is based on PBFT. Further, in section V, we describe the implemented simulator and the simulation parameters. Afterward, in section VI, we assess the performance of our work through simulations. Finally, section VII concludes this paper and discusses future works.

II. The State of Art

Using the blockchain to deal with the various challenges of VANETs has recently received a significant attention. To start with, many blockchain solutions have been designed using the PoW consensus protocol. The PoW was firstly introduced with Bitcoin [5], the first blockchain application; it consists of dedicated processors trying to solve a difficult but easy to verify cryptographic puzzle. The PoW is known in the literature for its high security since it cannot be compromised unless more than 50% of the system's total computational power are controlled by a malicious node [6]. That security is at the expense of high resource consumption due to the concurrent puzzle solving, and low throughput and high latency.

In the context of VANETs, the PoW has been used to build a distributed key management in heterogeneous intelligent transportation systems [7]. Also, in [8], basic concepts of blockchain were used for data sharing among intelligent vehicles while relying on the PoW algorithm. Besides, in [9], a blockchain of reputation was proposed for decentralized trust management for vehicular networks. That blockchain relies on a combination between PoW and Proof of Stake (PoS) [10] mechanisms for the consensus. Also, in [11], a blockchain featured with a PoW consensus was introduced for distributed trust management, without revealing vehicles identity. Similarly, in [12], the authors proposed a blockchain-based scheme for distributed trust management, where a fixed number of RSUs validate events using fuzzy logic and participate in the block creation process using PoW. Furthermore, in [13], the PoW mechanism was adopted for a blockchain-based secure, and trustworthy event message delivery in VANETs. The proposed architecture uses a zoning technique, where multiple blockchains work locally and independently based on the geographical location. As theoretically argued, the works mentioned above inherit the scalability issues of the PoW mechanism. They suffer from a high latency when it comes to validating transactions. As a result, such approaches are not suitable for our use case that comprises an important number of traffic events that must be written in the blockchain every second. Some works such as [14] have tried to reduce the PoW latency by speeding up the block interval; however, doing so might raise critical security issues and involve extra communication load due to potential forks.

On the other hand, other approaches have been built using consortium blockchains to deal with VANETs challenges. In a consortium blockchain, a predefined subgroup in the network agrees on the next block; that is usually based on the Practical Byzantine Fault Tolerance (PBFT) algorithm rather than the PoW.For instance, in [15], the authors leveraged a consortium blockchain for secure data sharing in vehicular edge computing; they presented a new communication model for the consensus process. The protocol seems simplified; however, further details regarding their security model are needed. Moreover, in [16], authors presented a consortium blockchain for secure data sharing and storage in VANETs based on PBFT [17]. Similarly, in another work [18], authors have proposed a consortium blockchain framework for traffic-related events management, building on the PBFT algorithm. Although the above mentioned works propose strong theoretical schemes based on PBFT, more performance studies are needed regarding the throughput and latency, as well as, the size of the blockchain when real world VANETs settings are considered. Adding to the performance, the literature lacks important details about how the PBFT can be adapted to validate and track traffic events in the transportation system. Furthermore, we believe that the PBFT can be improved more by dynamically selecting the

nodes that will participate in the consensus, as well as, using micro-transactions instead of full transactions.

Other works in the literature focused on building new consensus protocols dedicated for the VANETs context. Typical example are Proof of Event (PoE) [19] and Proof of Driving (PoD) [8]. However, further discussions are needed regarding their practical performance and security models. Such protocols do not often support malicious nodes; they are crash tolerant rather than fault tolerant such as the PBFT and PoW. As such, they sacrifice the security and resilience to achieve higher throughput and low latency.

III. VANETs Architecture

In this section, we give a brief description of the proposed architecture. We present the VANETs components and discuss their roles and interactions in the context of proposing a secure and trustworthy traffic-related events sharing system.

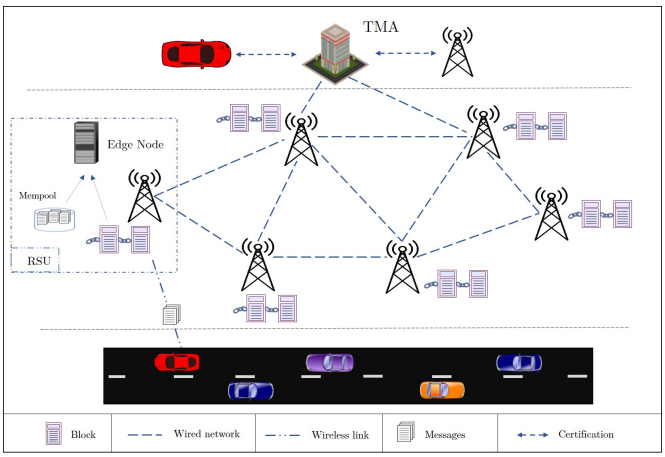

Fig. 1: System Architecture

The system architecture in VANET's can be separated into three layers as illustrated in Figure 1: the Ad-hoc layer, the infrastructures or Road Side Units (RSUs) layer, and the authority layer, which include the Traffic Management Authority (TMA). The first layer consists of intelligent vehicles communicating and exchanging data with other vehicles through their dedicated WAVE (wireless access in vehicular environments) modules. And through OBUs and dedicated sensors, vehicles collect and upload relevant traffic events (e.g., accidents, safety warning messages, traffic and weather condition announcements, and any hazardous events related to the traffic system). The communications reliability and authenticity are ensured through cryptographic encryption of exchanged messages. The collected data play a crucial role in the efficiency of the transportation system. Besides advertising traffic-related announcements, vehicles also receive information and safety warnings from RSUs and various services related to traffic system facilities, such as suggestions on the optimized trips and free parking places, etc. Vehicles' collaboration and engagement in uploading relevant and pertinent traffic records will contribute to improving traffic conditions; as a reward, they may be prioritized for the transportation system provided services as an incentive to their participation in providing crucial information. Doing so will motivate vehicles to upload relevant traffic-related data to RSUs. Details about such an incentive system are not in the scope of the paper. On the other hand, vehicles can decide to act maliciously by spreading fraudulent messages in the network. As such, recording vehicle communications with RSUs using a secure, immutable, transparent, and resilient ledger is a must.

The second layer in the VANETs architecture represents the infrastructures spread along at the roads' sides, i.e., Road Side Units (RSUs) to support vehicles with services. Each RSU is connected with nearby RSUs by a wired link, thus forming a peer-to-peer network. And each RSU is embedded with an edge node providing the required storage and computation to main a blockchain. Like vehicles, RSUs are equipped with wireless communication modules that help them gather messages from cars and to send warnings to these latters. We rely on a blockchain to ensure the correctness, transparency, and immutability of traffic-related messages sent by vehicles. The blockchain is distributed and maintained between the RSUs. Only RSUs participate in the consensus trying to validate traffic events and keep a trustful copy of the ledger. Before the consensus starts, RSUs have to collect messages from vehicles to validate a given traffic-related event. Collected records are stored independently by each RSU in a local database called *mempool*. Statements providing the same information are grouped to form one event; however, the identities of vehicles issuing the messages are also included in the event in order to track the complete, verifiable history of vehicle behavior. Each RSU thus makes a first local assessment of the trustworthiness of the received messages. Such local information will then pass through another verification step based on the PBFT consensus protocol before being added to the ledger.

The last layer in the vehicular network is the Trust Management Authority (TMA); its role in the system is to register the network participants by generating cryptographic credentials to newly joining vehicles and RSUs. Provided certificates must be anonymous, i.e., unlinkable to real vehicle identity to ensure vehicles' privacy.

IV. Proposed Protocol

In this section, we present the details of the PBFT-based consensus protocol that is used to track the traffic information in VANET's. We also explain the optimization techniques that are used to enhance the performance of that protocol.

Castro and Liskov first proposed PBFT in 1999 [17] as the first state machine replication algorithm to support Byzantine faults (i.e., irrational behaviors). The algorithm works on rounds; for each round, a replica (i.e., node) is

elected as a leader (primary), and other replicas act as backups. If the leader is compromised, the backups impose a "change view" that will replace the current leader with one of the backups. The "change view" process is necessary to ensure that the system progresses. Traditionally, PBFT relies on round-robin for the primary node election. However, such a design is prone to DDoS attacks since the next leader is known in advance. To cope with such a problem, the leader's election algorithm must be performed randomly by using a PoW protocol, for example. As will be shown later in the experimental results, using PoW for leader election will decrease the proposed protocol's overall performance, although it enhances its resilience.

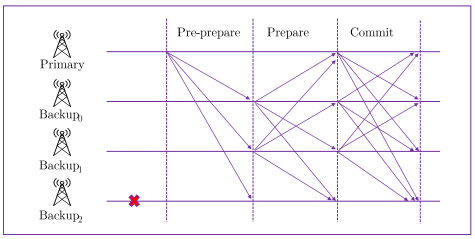

Fig. 2: PBFT communication workflow [17]

The PBFT algorithm guarantees both liveness and safety under the following three assumptions [17]:

1) Bound on faults: faulty replicas don't exceed $F = (n-1)/3$ over a lifetime of the system, where n is the total number of the consensus nodes.
2) Strong cryptography: consensus nodes are unable to subvert the adopted cryptographic techniques.
3) Weak synchrony: exchanged messages can't be delayed more than an asymptotic upper bound t, while t is a configurable parameter.

In PBFT, the nodes execute three communication phases ($Pre-prepare$, $Prepare$, and $Commit$) to reach a consensus on the next block as illustrated in Figure 2. Therefore, to guarantee the authenticity of messages, consensus participants' identities must be known in advance. Initially, in $Pre-prepare$, the leader multicasts a $pre-prepare$ message to the backup nodes. After receiving a $pre-prepare$ message, a backup node broadcasts a *prepare* message to all nodes during the $Prepare$ phase, including the leader; this phase guarantees that the leader has sent the same *prepare* message to the correct backups. After receiving $2F+1$ *prepare* messages that match the $pre-prepare$ message and from different nodes, a node will broadcast a *commit* message; thus, it moves to the $Commit$ phase. Like the $Prepare$ phase, after receiving $2F+1$ commits that match the $pre-prepare$ message, all correct nodes will append the block to their local copy of the blockchain.

As can be seen from Figure 3, the proposed consensus works like PBFT. After the selection of a RSU leader to forge the next block, three communication rounds $Pre-prepare$, $Prepare$, and $Commit$ are performed to agree on the next block of traffic events. The elected RSU selects a set of traffic events from its local *mempool* and forges a block containing the corresponding transactions; the leader then forwards the block to its neighbor RSUs. Those latter will act as validators for the forged block. Indeed, in VANETs, RSUs that are close to each other are likely to receive the same traffic events from different vehicles. Neighbor RSUs are therefore essential to assess the trustworthiness of traffic events that occur in their geographical scope. By doing so, we avoid submitting the forged block to all RSUs in the system. As a consequence, the communications load will be minimized, and the trustworthiness of messages will increase since only relevant RSUs will assess the validity of the traffic events. Importantly, to identify the nearest traffic events among all received messages, RSUs rely on the power of the signals received from the vehicles communications.

Upon receipt of a proposal (i.e., a batch of events to be confirmed), a validator RSU will verify the events relying on its local *mempool*. Then, the RSU will send a *prepare* message to all the other consensus nodes ($Prepare$). New proposal validation includes verifying its creator's digital signature, cryptographically ensuring the correctness of traffic-related transactions, and assessing its content. In other words, after receiving a block, the RSU checks the accuracy of all traffic events it has included.

For example, for a traffic-related event to be deemed valid, the block proposer must prove the reception of a given number of messages from different vehicles confirming the event's occurrence. To do this, he must include in the transaction the identities (and signatures) of all vehicles that have sent messages related to this event. The number of vehicle identity to be checked is a parameter. The use of a high number is essential to ensure a high level of event trustworthiness.

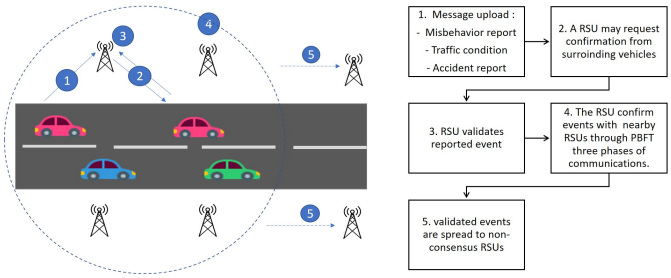

Fig. 3: An overview of the proposed solution

It is interesting to note that a RSU can act maliciously by not including all signatures in the transaction. For example, an RSU may only include signatures from vehicles confirming the occurrence of an event and ignore all messages from vehicles confirming the opposite. To detect that, when receiving the block, the receiver will also compare the validity of the events against its local *mempool* of transactions. We assume in this work that neighboring RSUs have relatively synchronized *mempools* that contain all traffic events flowing in their environment.

After receiving 2/3 *prepare* messages from all dynamically selected validator nodes (i.e., the nearest RSUs), the RSU broadcasts a *commit* message to all validator RSUs (*commit*). For a block to be validated and therefore added to the validator's blockchain, it must receive 2/3 validation messages from the validator nodes.

As can be noticed, two rounds of voting on a block are necessary to reach a consensus in our protocol. One is done during the *prepare* phase, and the other during the *commit* phase. Such votes are interestingly made between a dynamic number of RSUs that are close to each other, and that have enough knowledge to vote on a particular traffic event. After reaching consensus on the block, the block becomes publicly verifiable by other RSUs (non-consensus RSUs) thanks to cryptography signatures.

It is essential to mention that the traditional PBFT consensus protocol suffers from communication overhead: $O(n^2)$ in a typical scenario and $O(n^3)$ for the *change view* (i.e. in a compromised environment) , where n is the number of nodes participating in the consensus. In addition, in the traditional PBFT approach, all the RSUs in the system will hold a copy of the whole blockchain ledger. That will certainly results in a huge amount of communications, which may prevent the validator nodes from focusing on their validation tasks, as well as, an enormous amount of data that will be replicated among all nodes in the system. To deal with such challenges, we propose in this work to use micro-transactions instead of full transactions. Essentially, only the dynamic number of RSUs that will validate the traffic-events will store the resulting complete block. Other nodes will instead store blocks of micro-transactions.

We propose to truncate transactions to micro-transactions that contain only the data necessary to describe an event. For example, only the transaction hash, event type, and location could be stored in the micro-transaction. Micro-transactions are valid transactions and they contain sufficient data so that an RSU can evaluate its contents and decide whether to request the complete transaction or not. Ultimately, block replication is controlled, and an RSU may not store all the complete transactions in a given block. However, the addresses of complete block holders are defined in the micro-block, so the necessary events could be requested if required.

V. Simulator description

To validate the empirical performance of the proposed protocol while tracking traffic events, real-world settings have been simulated using NS-3 network simulator [20]. The proposed work has been implemented and linked to NS-3 as an external pluggable consensus. All the signatures of RSU validators and vehicles are based on Schnorr signature [21], and they have been implemented using the OpenSSL library.

To simulate a vehicular network, we distinguish between two types of nodes, RSU and Vehicle. RSUs are connected by a point-to-point channel, forming a peer-to-peer network. At the beginning of the simulation, we randomly connect each RSU to a minimum number of other RSUs; we make sure that RSUs network forms a connected graph; thus, we avoid network partitions. To easily manage communications between RSUs without being bothered by the application routing layer, each RSU maintains a TCP connection with its peers. Also, each vehicle is connected to a group of RSUs. Vehicles generate transactions according to Poisson distribution.

To simulate the connection between vehicles and RSUs, a wired link module is used. Furthermore, we picture the life cycle of generated transactions from reception, validation, and storage in the *mempool*, to being packed in a block, and finalized in the blockchain ledger. The Vehicles' main function is events generation, while RSUs focus on forging and validating the next block of traffic records to be appended to their local copy of the blockchain.

Our main focus is to assess the performance (i.e., throughput and latency), storage, and communication load of the proposed protocol with the increasing consensus group size (k). For this reason, we have fixed some parameters such as the block size (i.e., the maximum number of events in a block) to 1000 events; the total number of RSUs to 20; the network speed to 100 Mbps; and the time to wait for events λ_{txs} to 500 ms; events generation time is fixed to 4s; and event size to 800 *bytes* as defined in [9, 16]. In table I, we summarize our simulator parameters.

Description	**Value(s)**
running consensus	enhanced PBFT
# of RSUs	20
k : consensus groupe size	$[4, 7, 10, 13, 16, 19, 20]$
f : events generation rate	2000 event/s
event/transaction size	800 *bytes*
micro-transaction size	300 *bytes*
max # of events per block	1000
p2p link speed	100 Mbps
p2p link latency	$1ms$
λ_{txs} : time to wait for events	$500ms$

TABLE I: Simulator parameters

The parameters above include real-world configurations. For instance, 20 RSUs can cover a whole city when supposing RSUs communicate using WIMAX [22] which has a coverage rate of 15 km in an urban environment and bandwidth of 100 Mbps. Regarding the event arrival frequency, we set it high enough 2000 event/s to capture heavy traffics and the expected increase in the number of autonomous vehicles in the short future.

VI. Evaluation

In this section, we simulate various instances of the implemented protocols using a server with the flowing settings: Dell R640 server, Intel(R) Xeon(R) Silver 4112; CPU 2.60GHz; 8 core CPU; 64 GB RAM, and running

Ubuntu 18.04. The assessed metrics are throughput, latency, data storage, and communication loads. The results represent end-to-end measurements from all RSUs. The throughput is examined by dividing the total confirmed events by the consensus time, and, for transaction latency, we subtract the creation timestamp from the confirmation timestamp. Each experiment is repeated for 5× with different seeds, and the mean is plotted.

A. Protocol performance

To study our protocol's scalability with the increasing size of the dynamic consensus group (k), we run experiments where we vary k from 4 to 20 with a step of 3; also, we fix f to 2000 event/s and then assess the throughput and latency of the blockchain. Such configuration implies that 2000 vehicles exist at the same time in the same region, and each vehicle is generating one traffic event per second and sends it to the k selected RSUs that will validate and consequently reach a consensus on the data state. Figure 4 shows the results of this experiment with our protocol tested in two different scenarios. A first scenario where a PoW (PBFT-PoW) is used for leader election and a second scenario based on a Round-robin (PBFT-Robin) mechanism for electing the leader.

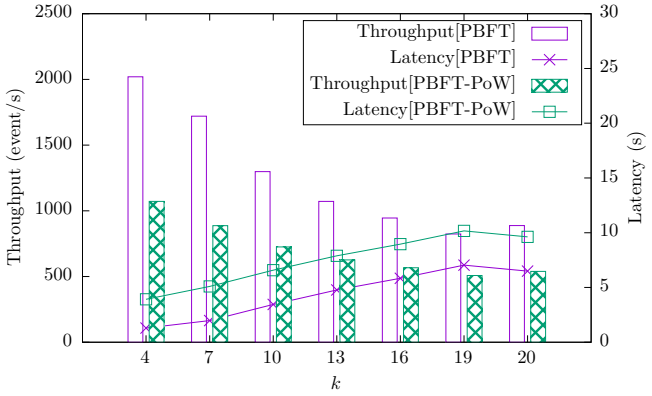

Fig. 4: Performance (i.e., throughput and latency) vs. k (consensus group size)

As can be seen from Figure 4 , results indicate that increasing k drastically affects the overall performance of our protocol. That is because the proposed protocol lies in the PBFT quadratic ($O(k^2)$) communication cost with the size of consensus participants. In the worst case, when k is set to 19, the latency may increase to $10s$ for PBFT-PoW and $7s$ for PBFT-Robin. Besides, it is worse mentioning increasing the value of k will also increase the security of the consensus, as well as, the trustworthiness of messages stored in the blockchain. Thus, a trade-off is faced between the security, trustworthiness and the performance (i.e., throughput and latency). Moreover, we can notice that if $k = 20$, the performance is slightly better than when $k = 19$. In fact, unlike when $k = 19$, if $k = 20$, there is no need to spread the block to RSUs that did not participate in the consensus. Results also show that adopting the PoW in the leader election induces around $3s$ on the latency compared to PBFT-Robin. Besides, in terms of throughput, PBFT-Robin outperforms PBFT-PoW; for instance, if $k = 7$, PBFT-Robin' throughput is $\approx 1.6\times$ better than the PBFT-PoW's. However, as previously mentioned, the round-robin election is more prompt to DDoS attacks on the leader, causing leader change, thus slowing down the consensus.

From the above results, it can be noticed that the proposed protocol is efficient enough to be used in VANETs when the number of the RSUs k is carefully selected. That is done in practice depending on the number of vehicles, the number of events generated by vehicles, as well as, the desired amount of security and resilience.

B. Effectiveness of micro-transactions

In this subsection, we aim at assessing the effectiveness of micro-transactions regarding throughput and latency. In Figure 5, we plot two setups of our protocol; in the first configuration, the full blockchain is replicated among all the RSUs (FULL) and in the other to only RSUs that participated to the consensus. Non-consensus RSUs will store just micro-blocks as previously mentioned. We vary the number of RSUs participating in the consensus (k) and we don't limit the block size since it is the optimal setting when $f = 2000$ event/s. We also fix the network latency to $10ms$.

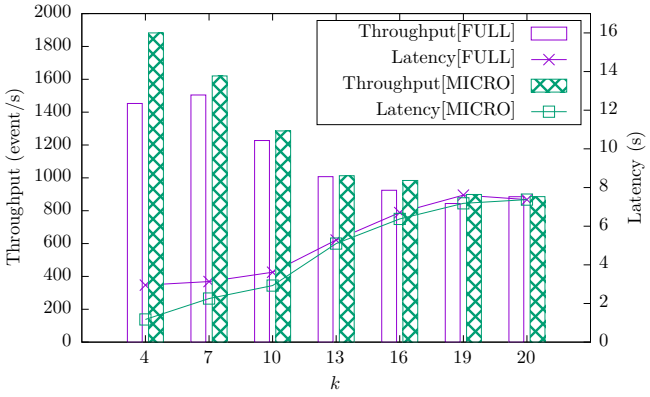

Fig. 5: The impact of micro-transactions on the replication (k), network latency = $10ms$

As can be seen from Figure 5, results show that when k is equal to 4 among 20 RSUs, the throughout of the "FULL" version is less than the "MICRO". That is because less communications are made between the validator RSUs when micro-blocks are exchanged. Moreover, less communications are made in the whole network making the validator nodes only focus on the validation process rather than wasting their time in broadcasting blocks to non-validator RSU. Additionally, results indicate the same performance for both settings ("MICRO" and "FULL") when $k = 20$; that is because $k = 20$ means that all

RSUs participate in the consensus. Therefore, there are no micro-transactions to transmit to non-consensus RSUs; thus, "MICRO" is the same as "FULL."

Furthermore, results show an overall decrease in the performance of the "MICRO" setting with the increasing of k; this is because increasing the number of RSUs to valid a block (k) decreases the number of non-consensus RSUs for every phase of the consensus. Hence, more full blocks are exchanged throughout the network; therefore, minimizing the effect of micro-transactions, and as a result, worsening the system's performance.

It can be noticed from the above results that using micro-blocks relatively enhance the overall performance of the proposed protocol. That optimised efficiency will certainly enlarge the spectre of the use cases where the proposed protocol can be applied. In addition, it allows for cost reduction since the communications are restrained to a small number of nodes. That becomes more important in a larger vehicular network.

C. Communication load and storage overhead

In Figure 6, and Figure 7, we assess the impact of micro-transactions on the communication load (i.e., the amount of exchanged data between RSUs) and the storage overhead, respectively. In both figures, we plot the two protocols, "Full replication", and "k-replication", aiming to show the latter's effectiveness regarding communication and storage load.

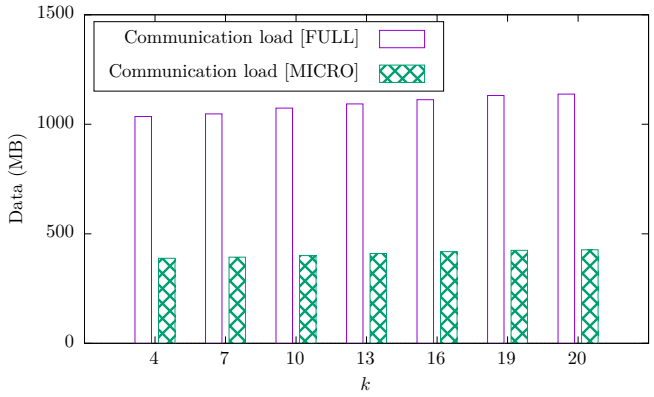

Fig. 6: The communication load vs. consensus group size - k

Figure 6 shows that the communication load increases with k. Increasing k from 4 to 20 increases the amount of data exchanged between RSUs from 388 MB to 426 MB (38 MB) and 1034 MB to 1138 MB (104 MB) for respectively when the protocol is set to micro-transactions and the full-replication. This is because a higher k requires more communications during the consensus phase ($O(k^2)$). Surprisingly, the communication load increases only slightly with the increasing k; this is because RSUs multicast blocks directly to their peers, without asking if they have already received the same block, aiming to speed up the time to spread a block throughout the network. Hence, the high communication cost. Results also show that using micro-transactions minimizes the communication costs; that was expected since micro-blocks are transmitted to non-consensus participants $(n - k)$ instead of a full block. On average, the communication load is $\sim 2.6\times$ lesser when using micro-transactions, which corresponds to the ratio between transaction and micro-transaction sizes.

Figure 7 measures the blockchain size with the increasing consensus participants number (k). The plotted results represent the average size in MB of the local copy of the blockchain stored by the RSUs.

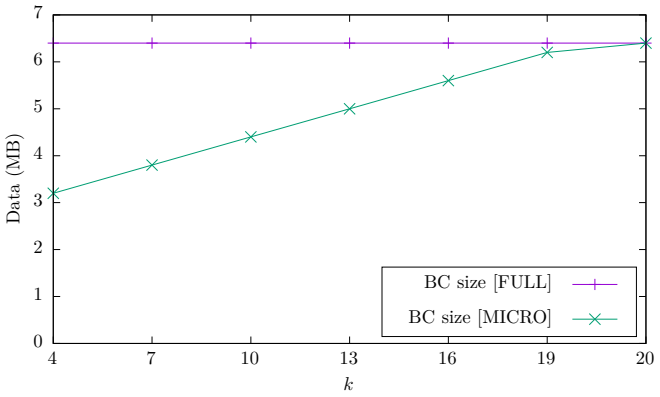

Fig. 7: The storage cost vs. consensus group size - k

The size of the blockchain is calculated by this formula : BC size $= (n - k) * (bs_{micro}) + k * (bs)$, where bs_{micro} indicates the average size of the micro-blocks. As can be seen, when the consensus group size (k) increases, the blockchain size increases. This increase is due to the increase in the number of nodes storing complete blocks resulting from the increase in the number of nodes involved in the consensus. For example, by increasing k from 4 to 20, the blockchain's size increases from $\sim$ 3.2 MB to $\sim$ 6.4 MB (doubling the size of the blockchain). The results also show that if $k = 20$, the blockchain's size becomes equal to the full duplication. That is because all the nodes have become consensus nodes; thus, they store a complete copy of the blockchain rather than partial data through truncated events (i.e., micro-transactions). On the other hand, for full replication, the block chain's size (BC size) remains invariant because blocks are replicated among all RSUs. We wait until all generated transactions are finalized before measuring the size of the blockchain.

VII. Conclusion And Perspectives

This paper presented a new blockchain based protocol for securing traffic messages in VANETs. The RSUs are used as blockchain nodes; they receive traffic events from vehicles and validate their trustworthiness using an improved version of the PBFT protocol. The proposed algorithm uses a dynamic selection of RSUs to participate

in the consensus, and micro-transactions which are light copies of the original traffic events records. The selection of consensus participants is dynamic; it is based on the proximity of RSUs from the traffic events themselves. Furthermore, we presented in this paper a simulation framework built on top of NS-3, whereby we evaluate different instances of the studied protocol and assess several metrics such as the throughput, latency, storage, and the communication loads of the blockchain. The experimental results show that the proposed protocol outperforms the traditional PBFT from a performance point of view. In addition, using micro-transactions helps in increasing the blockchain throughput, as well as, reducing the blockchain size and the amount of communications in the network. In future works, we have planned to introduce a credibility scoring module, seeking to enhance the selection of RSUs; not all RSUs around will be used; only the most credible and nearest ones. We also endeavour to study the impact of the mobility of vehicles on the performance of the proposed protocol. Finally, we also project to investigate different solutions to mitigate the PBFT single leader vulnerability, always in the VANETs' context.

References

[1] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[2] R. Mishra, A. Singh, and R. Kumar, "Vanet security: Issues, challenges and solutions," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, 2016, pp. 1050–1055.

[3] X. Li, Z. Zheng, and H.-N. Dai, "When services computing meets blockchain: Challenges and opportunities," *Journal of Parallel and Distributed Computing*, vol. 150, pp. 1–14, 2021.

[4] V. Astarita, V. P. Giofrè, G. Mirabelli, and V. Solina, "A review of blockchain-based systems in transportation," *Information*, vol. 11, no. 1, p. 21, 2020.

[5] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[6] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International workshop on open problems in network security*. Springer, 2015, pp. 112–125.

[7] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.

[8] M. Singh and S. Kim, "Blockchain based intelligent vehicle data sharing framework," *arXiv preprint arXiv:1708.09721*, 2017.

[9] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[10] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.

[11] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for vanets," *IEEE Access*, vol. 6, pp. 45 655–45 664, 2018.

[12] A. Kchaou, R. Abassi, and S. Guemara, "Toward a distributed trust management scheme for vanet," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 2018, p. 53.

[13] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new-type of blockchain for secure message exchange in vanet," *Digital Communications and Networks*, 2019.

[14] M. Wagner and B. McMillin, "Cyber-physical transactions: A method for securing vanets with blockchains," in *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2018, pp. 64–73.

[15] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, no. 5, 2018.

[16] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular adhoc network," *IEEE Access*, vol. PP, pp. 1–1, 01 2019.

[17] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[18] E.-h. Diallo, K. Al Agha, O. Dib, A. Laube, and H. Mohamed-Babou, "Toward scalable blockchain for data management in vanets," in *Workshops of the International Conference on Advanced Information Networking and Applications*. Springer, 2020, pp. 233–244.

[19] Y. Yang, L. Chou, C. Tseng, F. Tseng, and C. Liu, "Blockchain-based traffic event validation and trust verification for vanets," *IEEE Access*, vol. 7, pp. 30 868–30 877, 2019.

[20] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[21] C. P. Schnorr, "Method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system," Feb. 19 1991, uS Patent 4,995,082.

[22] S. J. Vaughan-Nichols, "Achieving wireless broadband with wimax," *IEEE computer*, vol. 37, no. 6, pp. 10–13, 2004.