Study and Design of Blockchain-based Decentralized Road Traffic Data Management in VANET (Vehicular Ad hoc NETworks)

El-Hacen Diallo

université PARIS-SACLAY

Study and Design of Blockchain-based Decentralized Road Traffic Data Management in VANET (Vehicular Ad hoc NETworks)

Étude et conception d'une gestion décentralisée des données du trafic routier basée sur la Blockchain dans un réseau VANET (réseaux ad hoc véhicules)

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Réseaux, information et communications
Graduate School : Informatique et sciences du numérique, Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche Laboratoire interdisciplinaire des sciences du numérique (Université Paris-Saclay, CNRS), sous la direction de Khaldoun AL AGHA, Professeur

Thèse soutenue à Paris-Saclay, le 1 Avril 2022, par

El-hacen DIALLO

Composition du jury

Pascal URIEN Professeur, Télécom Paris	Président
Houda LABOID Ingénieure, entreprise Huawei	Rapporteure & Examinatrice
Maria POTOP-BUTUCARU Professeur, Sorbonne Université, laboratoire LIP6	Rapporteure
Guy PUJOLLE Professeur émérite, Sorbonne Université, laboratoire LIP6	Examinateur
Nadjib AIT SAADI Professeur, Université de Versailles-Saint-Quentin-en-Yvelines, laboratoire David	Examinateur
Khaldoun AL AGHA Professeur, Université Paris-Saclay	Directeur de thèse

THESE DE DOCTORAT

NNT : 2022UPASG017

Study and Design of Blockchain-based Decentralized Road Traffic Data Management in VANET (Vehicular Ad hoc NETworks)

El-hacen DIALLO

Abstract

The prominence of autonomous vehicles has imposed the need for more secure road traffic data (i.e., events related to accidents, traffic state, attack report, etc.) management in VANET (Vehicular Ad hoc NETworks). Traditional centralized systems address this need by leveraging remote servers far from the vehicles. That is not an optimal solution as road traffic data must be distributed and securely cached close to cars to enhance performance and reduce bandwidth overhead. Blockchain technology offers a promising solution thanks to its decentralization property. But some questions remain unanswered: how to design blockchain-adapted traffic data validation, which is more complex than an economic transaction? What is the performance in real-world VANET scenarios?

This thesis addresses those questions by designing blockchain-adapted traffic data management. The performance analysis and the validation of the proposed schemes are conducted through various simulations of real scenarios.

We first adapt the PoW (Proof of Work) consensus mechanism to the VANET context whereby the RSUs (Road Side Units) maintain the decentralized database of road traffic data. After that, the proposed scheme is evaluated in the presence of malicious vehicles. The results show that the proposed approach enables a secure and decentralized database of road traffic data at the RSUs level.

Next, motivated by our findings, we adopt PBFT (Practical Byzantine Fault Tolerance), a voting-based consensus mechanism, to reduce the blockchain latency. The traffic data validators are dynamically selected based on traffic event appearance location. Finally, we propose a novel blockchain replication scheme between RSUs. This scheme offers a trade-off between the blockchain latency and replication frequency. Simulation results show better performance when the validators (i.e., RSUs) are minimized.

Finally, we propose a trust model to minimize the validators without compromising the decentralization and fairness of block-creation. This trust model leverages the geographical distance and the RSUs trust to dynamically form a group of validators for each block in the blockchain. We formalize and evaluate this trust model, considering various scenarios with malicious RSUs. Results show the efficiency of the proposed model to minimize the validators group while isolating malicious RSUs.

Synthèse

La sécurité routière est un enjeu social majeur depuis de nombreuses années. Les progrès technologiques dans les domaines des communications et des systèmes d'information favorisent le déploiement des Systèmes de Transport Intelligents (STI). Les STI sont censés optimiser les services de transport ainsi que la gestion et la sécurité du trafic routier. Cependant, la question de la cybersécurité dans les STI reste un défi majeur. Les solutions de sécurité existantes continueront d'évoluer pour faire face à l'évolution des attaques et des vulnérabilités, mais les problèmes critiques nécessitent encore la définition de nouvelles solutions innovantes et efficaces.

Le développent des technologies sans fils a accentué les capacités de communication et de connectivité des véhicules, avec des infrastructures routières, avec des infrastructures télécoms et avec d'autres véhicules. On parle alors de systèmes hybrides véhiculaires dont une partie est constituée d'un réseau autonome appelé VANET (réseaux ad hoc véhicules), qui est le composant central des STI. Au cours des deux dernières décennies, les académiciens ainsi que les industriels automobiles se sont intéressés sur plusieurs applications VANET pour la sécurité et l'optimisation du transport.

En outre, la maturité des technologies véhiculaires notamment des logiciels et capteurs embarqués permettront aux véhicules du futur de collecter et de partager des données sur l'état du trafic routier. Ces données pourront être utilisées, pour proposer des recommandations d'itinéraires optimisés, permettant de réduire la consommation de carburant, les émissions de CO2, et les accidents. Toutefois, il est primordial de sécuriser les données en amont pour que ces dernières puissent être exploiter sans crainte. Plus précisément, plusieurs exigences de sécurité doivent être satisfaites, notamment le partage et le stockage des données fiables ainsi que la garantie de leur disponibilité sans oublier la protection de la vie privée des véhicules.

La blockchain grâce à ses caractéristiques de décentralisation, d'utilisation de la cryptographie, d'auditabilité, de transparence et d'immuabilité apparait comme une technologie intéressante à être explorée dans le cadre des Systèmes de Transport Intelligents. En effet, elle permet principalement de stocker et d'échanger l'information de manière sécurisée et transparente, sans avoir besoin de recourir à un organe de contrôle ou un tiers de confiance.

Les travaux de la thèse rapportés ici, se sont focalisés particulièrement à la conception et à l'étude d'une gestion décentralisée des évènements du trafic routier au moyen de la technologie blockchain. Ces travaux prennent en compte les con-

traintes de transparence, de décentralisation, et de la robustesse en termes de sécurité tout en considérant les caractéristiques des données échangées dans le réseau VANET et les exigences de ces applications.

La thèse est organisée en huit chapitres. Le premier introduit le contexte général de la thèse en décrivant les différentes facettes liées à la problématique traitée, qui est la sécurité et la gestion des données du trafic routier.

Le chapitre 2 présente un aperçu sur les VANETs ainsi que sur la technologie blockchain. Il vise à positionner les problèmes de recherche et dresse un état de l'art sur les VANETs ainsi que sur la technologie blockchain. Le chapitre débute par une description des modèles d'architectures associés aux VANETs, leurs caractéristiques et les problèmes de sécurité ciblant aujourd'hui ces systèmes. La deuxième partie introduit la technologie blockchain à travers ses propriétés, ses différents types, les applications supportées, ses principaux mécanismes et les défis associés. A travers ce chapitre, on motive l'exploration de l'intégration de la technologie blockchain dans les réseaux véhiculaires.

Le chapitre 3 dresse une analyse de l'intégration de la technologie blockchain dans le domaine des réseaux véhiculaires en mettant en exergue ses limites vis-à-vis de la problématique de la sécurisation de la gestion des données du trafic routier. Trois modèles d'architectures ont été ainsi analysés : architectures centralisées, architectures distribuées et architectures décentralisées ou basé sur la blockchain. Nous analysons et soulignons les avantages et les inconvénients de chaque modèle. Aussi, nous justifions le choix d'adopter l'approche décentralisée en ciblant comme objectifs d'améliorer la sécurité, de traiter le passage à l'échelle et de fournir une analyse de performances plus avancée.

Le chapitre 4 est consacré à la première contribution. Elle commence par étudier l'utilisation d'une blockchain de type Bitcoin pour sécuriser le stockage des données de trafic. L'algorithme PoW (preuve de travail) est utilisé comme mécanisme de consensus. Différentes étapes ont été analysé afin d'évaluer la faisabilité de cette intégration en commençant par resituer le modèle architectural composé de trois couches. Il s'agit de la :

- couche ad hoc constituée de véhicules interconnectés entre eux via des communications sans fil (IEEE 802.11p ou ITS-G5);
- couche constituée par les unités bord de route (RSUs) interconnectées entre elles et
- couche constituée par le centre de gestion de trafic routier appartenant à des opérateurs routiers publics ou privés.

Les idées avancées dans ce chapitre ont été implémentées et testées à travers le simulateur NS-3 en considérant des scénarios de trafic réel. Plusieurs métriques ont été utilisées pour évaluer les performances de la solution proposée, il s'agit du débit (nombre d'évènements par seconde), de la latence, de la difficulté de PoW en fonction de la variation du nombre de RSUs et le taux d'arrivée des évènements de trafic. Pour la sécurité et la fiabilité, celles-ci ont été évalué au moyen de taux de blocs invalides et de la taille de fork la plus longue.

Dans le chapitre 5, la robustesse de la solution proposée dans le chapitre précédent est étudiée vis-à-vis des comportements malveillants des véhicules. Ce chapitre s'est penché également sur la partie validation des évènements du trafic routier envoyés par les véhicules avant qu'ils ne soient enregistrés dans la blockchain et disséminés par la suite dans le réseau des RSUs. L'approche utilisée est basée sur l'utilisation d'un seuil. Une analyse par simulation a été réalisée, en se focalisant sur l'effet de faux évènements, sur le délai de confirmation d'un évènement, l'effet du seuil, sur le nombre de faux évènements, sur la latence et aussi le taux de blocs non valides en faisant varier le seuil.

Le chapitre 6 présente un nouveau protocole en changeant le mécanisme de consensus en l'occurrence PBFT (tolérance de panne byzantine pratique) qui réduirait la latence. Nous avons commencé par présenter les micro-transactions afin de contrôler la réplication des blocs, le modèle de données est ensuite fourni avant de présenter une description détaillée du protocole. L'idée clé de cette contribution est celle de regrouper dynamiquement les k RSUs basés sur la localisation des évènements pour améliorer la validation des évènements. Le paramètre k est considéré comme la taille du groupe de consensus. Le simulateur développé dans le chapitre 4 est utilisé pour réaliser l'implémentation et la validation par simulation du protocole en considérant un large volume d'évènements. Les métriques évaluées sont la latence, les surcharge de communication et de stockage en faisant varier les paramètres essentiels du protocole à savoir k, le taux d'arrivée des évènements, le schéma de réplication. Une comparaison avec certains protocoles de l'état de l'art a été également effectuée. Les résultats obtenus dans ce chapitre montrent de bonnes performances avec un nombre réduit de RSUs.

Le chapitre 7 se concentre sur la définition d'un modèle de confiance pour isoler les RSUs malicieux et maintenir le nombre de RSUs validateurs des événements réduit. Pour ce faire, nous proposons de combiner la notion de proximité des RSUs à l'évènement et la réputation des RSUs. L'approche a été validée par simulation.

Le chapitre 8 conclut la thèse et évoque les perspectives.

To my parents, of course.

Acknowledgement

Je remercie Dieu de m'avoir donné la force, l'opportunité de faire un doctorat, la persévérance d'enduire les péripéties d'un tel projet, ainsi que la détermination de le finir de manière satisfaisante.

Je tiens à remercier particulièrement Monsieur Khaldoun Al Agha, mon directeur de thèse, pour sa disponibilité et son soutien tout au long de la thèse. Je remercie également Monsieur Omar DIB pour son implication et son optimisme qui a contribué à forger ma confiance envers mes travaux.

Je tiens à remercier les membres du jury, et en particulier Madames Houda LABOID et Maria POTOP-BUTUCARU d'avoir accepté de rapporter mes travaux de thèse.

Mes remerciements spécialement à l'endroit de mes bien-aimés parents, des membres de ma famille, et de mes amis (es) pour m'avoir toujours soutenu. Vous avez tous une place spéciale dans mon cœur.

Je tiens aussi à exprimer ma reconnaissance envers l'Ecole Polytechnique de Nouakchott pour la formation de qualité reçue. Je remercie particulièrement Mohamed Aly Louly et Hafedh Mohamed Babou pour leur soutien inestimable.

Mes remerciements également à tout le personnel du Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), permanents, thésards, post-docs, stagiaires, et surtout l'équipe ROCS pour l'accueil cordial et pour les moments riches d'échange scientifique. Qu'ils veuillent trouver ici, l'expression de mon amitié.

Enfin, je fais une mention spéciale à la société British Petroleum (BP), pour m'avoir octroyé la bourse, qui m'a permis de réaliser mon doctorat.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AI Artificial Intelligence.

BFT Byzantine Fault Tolerance.

CC Cloud Computing.

DDoS Distributed Denial of Service attack.

DSRC Dedicated Short Range Communications.

ECC Edge Cloud Computing.

GPS Global Positioning System.

HVC Hybrid Vehicular Cloud.

IaaS Infrastructure-as-a-Service.

IoT Internet of Things.

ITS Intelligent Transportation Systems.

MANET Mobile Ad hoc NETwork.

MEC Mobile Edge Computing.

P2P Peer-to-Peer.

PaaS Platform-as-a-Service.

PBFT Practical Byzantine Fault Tolerance.

PoA Proof of Authority.

PoET Proof of Elapsed Time.

PoS Proof of Stake.

PoW Proof of Work.

QoS Quality of Service.

RSUs Road Side Units.

SaaS Software-as-a-Service.

SCP Stellar Consensus Protocol.

TA Trusted Authority.

TEE Trusted Execution Environment.

TMA Traffic Management Authority.

V-CC VANET-based Cloud Computing.

V2I Vehicle-to-Infrastructure.

V2V Vehicle-to-Vehicle.

V2X Vehicle-to-Anything.

VANET Vehicular Ad hoc NETworks.

VCC Vehicular Cloud Computing.

VEC Vehicular Edge Computing.

VN Vehicular Network.

VuC Vehicles using the Cloud.

WAVE Wireless Access in Vehicular Environments.

X2I Anything-to-Infrastructure.

XGS Intel Software Guard Extensions.

1 Introduction

Contents

1.1 Intelligent Transportation Systems

Transportation systems have undergone massive development over the past decade due to rapid industrialization and urbanization. This transformation is driven by trends in advanced communication technologies and *intelligence* integration into transportation models. The result improves safety and security in the transportation sector and makes its services more intelligent and convenient: this is Intelligent Transportation Systems (ITS). According to EU Directive 2010/40/EU (7 July 2010), ITS is expected to optimize services in all transportation applications, especially traffic management, security, safety, and monitoring applications.

In addition, besides safety, security, and the global transportation Quality of Service (QoS) enhancement, ITS is also part of the sustainable development. It promotes green car models and modes of transport to cope with the environmental and climate challenges. This last point is subject to intense economic competition worldwide. Furthermore, with Internet of Things (IoT) and Smart solutions integration, ITS is experiencing massive growth in the market. According to Grand

View Research. Inc, the ITS market size is estimated to be USD 37.64 billion by 2027 [1].

Driven by innovation, ITS is growing worldwide and has become the focus of policy and legislative initiatives in the USA and Europe. Consequently, ITS standards had been defined by the U.S. Department of Transportation (USDOT), the European standard ETSI, and the ISO TC204 standard to promote future ITS deployment. Among elements of these standards, wireless communication technologies, sensors, and Vehicular Ad hoc NETworks (VANET) architectures and applications play a critical role.

1.2 Context and Motivation

VANET as a central component of ITS, supports road traffic services development and human safety. Each year, about 1.35 million people die, and more than 20 million are injured on the roads [2]. In addition, traffic jams result in massive loss of time and fuel, which affects human health and pollutes the environment.

Over the past two decades, scientists, governments, and the automotive industry have given attention to VANET to propose several applications for road safety and traffic efficiency. VANET provides Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communications that enable vehicles to share information that can be used to enhance human safety, optimize trips plans, and reduce CO2 emissions.

Before VANET can be deployed for real applications, its enabling technologies must be well studied and standardized. Many vehicular network projects and consortia were dedicated to that concern. For example, VSC and CAMP/VSC-2 were launched in the USA, SmartWay in Japan, Car to Car Communication Consortium (C2C-CC) and COOPERS in Europe. They discuss VANET architectures, standards, and protocols while defining its applications and challenges. When designing VANET applications, distinctive characteristics such as vehicle mobility and rapid topology change are the first concerns. But also, road traffic data security and reliability are among the major challenges in improving road safety and efficiency.

With the maturity of sensor and wireless communication technologies, vehicles are becoming more intelligent. Modern vehicles are equipped with sophisticated devices, sensors, and communication modules, enabling them to sense and share traffic data (e.g., road state). This data can be used, for instance, to propose optimized route recommendations. Traffic jams thus can be minimized, reducing fuel consumption, CO2 emissions, and accidents resulting from traffic jams. However,

the traffic data must be secured first; specific security requirements must be met. Secure sharing and storage are the main concerns [3–6]. A reliable and trustworthy database of traffic data is needed to feed VANET applications. This database must ensure data integrity, traceability, availability, and vehicle privacy [6, 7].

Existing traditional architectures store and manage traffic data relying on a trusted central cloud [8]. However, such a centralized architecture is prone to privacy issues high bandwidth consumption, leading to severe network congestion [9] that will increase data access delay. Because of this increase in delay, centralized architecture cannot meet the required QoS when there are a huge number of requests from vehicles [10]. Furthermore, centralized architectures are subject to single-point-of-failure issues [11, 12].

The Edge Cloud Computing (ECC) paradigm was proposed to mitigate the centralized architecture performance limitations. It consists of splitting the cloud and bringing it closer to the end devices [9]. Bringing facilities closer to vehicles reduces network congestion and bandwidth usage. As a result, vehicular data processing delay will be minimized, and VANET applications QoS will be enhanced. However, although ECC attenuates the latency issues in the centralized architecture, it raises further security concerns. Because, unlike the central cloud, which is equipped with sophisticated security mechanisms, Edge facilities are widely spread and are more vulnerable to physical attack [13, 14]. Consequently, maintaining a consistent database at the Edge Cloud level is a challenge by itself [14]. In other words, ensuring a transparent state between widely distributed Edge devices while remaining secure against attacks is a must.

Because of its exciting properties such as decentralization, immutability, and high fault tolerance, blockchain appears as an exciting technology to be investigated for traffic data management. This technology has recently attracted much attention from both scientists and industrials and has found use cases in a large set of domains [15], in the last few years.

Essentially, blockchain consists of a peer-to-peer network coordinated by a consensus algorithm to agree on the state of a distributed ledger. From a data structure perspective, a blockchain is a linked list of blocks, where each block stores the previous block's hash. Blockchain technology has emerged with revolutionary promises towards more secure applications design due to its exciting properties. This technology by design offers decentralization of the network, transparency of all actions, and immutability of data once written in the ledger. It also ensures high robustness in an untrusted environment without relying on a single central authority [16]. These properties stem from the consensus mechanism orchestrated between all or parts of the blockchain network.

Recently, blockchain technology was relied upon to build a secure and reliable

database of traffic data [15, 17]. However, the adaptability of this technology to road traffic data management is not yet mature; further study of this technology for VANET applications is needed. Several aspects need to be considered in that journey, including the VANET network specificities and VANET applications security and performance requirements.

1.3 Problem Statement

This thesis addresses the following question:

How to build a secure history of road traffic data in VANET, while data integrity and reliability are guaranteed within reasonable delays?

We decompose this research problem into three sub-problems:

- First, How to build a distributed and decentralized database of road traffic messages while ensuring data integrity, availability, and transparency?
- Second, given the increase in cyber-attacks, how to build a system with high fault tolerance and robustness against attacking vehicles and RSUs?
- Finally, how to respond to the above requirements in a reasonable delay to meet the time-critical nature of VANET applications and their QoS requirements?

1.4 Contributions

The main contributions of this thesis are summarized as follows :

1. We conduct a comprehensive literature review on traffic data management systems while describing their properties. In addition, we present an in-depth study of existing blockchain-based approaches for secure road events (e.g., accidents, traffic state, attack reports, etc.) management. Finally, we position our work based on the current state-of-the-art.
2. We propose a blockchain-based architecture for traffic event management. The aim is to integrate blockchain into VANET without compromising vehicle tasks such as broadcasting safety-critical announcements. The proposed

scheme is based on Proof of Work (PoW) consensus mechanism [18] orchestrated by the RSUs. Finally, we conduct a performance analysis of the proposed protocol through simulation; its effectiveness and robustness against attacking (i.e., malicious) vehicles are evaluated and discussed.

3. We propose a more scalable protocol to secure traffic messages in VANET. Instead of PoW, we rely on Practical Byzantine Fault Tolerance (PBFT) algorithm [19], a voting-based consensus mechanism, performed by the RSUs. The RSUs are dynamically selected based on their proximity to event occurrence location to ensure decentralization. Moreover, we introduce the concept of *micro-transactions* to minimize the consensus costs (i.e., computation, storage, and communication). Finally, we conduct an extensive performance study of the proposed protocol and evaluate the efficiency of micro-transactions. We also compare this contribution with comparative approaches in the state-of-the-art.

4. We propose a trust model to complement the previous contribution. The objective is to minimize the consensus group without compromising the decentralization of block-creation and the robustness of events validation. Therefore, besides RSUs proximity to the event location, we consider their reputation. Finally, the effectiveness of this proposal is evaluated under various scenarios of malicious RSUs.

5. We developed a blockchain simulator wherein we conduct extensive performance evaluations and simulate various configurations to validate our contributions. We consider complete scenarios where we capture vehicle mobility and V2I communications. We also simulate attacking vehicles and RSUs and study the robustness of the proposed protocols in handling such situations.

1.5 Thesis Organization

The thesis is organized into 8 chapters which are summarized below :

- Chapter 1 presents the context and research motivations of the thesis.
- Chapter 2 gives background knowledge on VANET and blockchain technology. It first provides an overview of VANET architecture, applications requirements, and challenges. It then introduces blockchain technology: its properties, applications, and challenges.

- Chapter 3 presents existing solutions for road traffic data management. It gives a comprehensive overview of existing architectures and discusses their design and limitations. Finally, the state-of-the-art gap is identified, and our contributions are positioned.
- Chapter 4 presents our first contribution, a blockchain-based architecture for secure storage of road traffic data. This contribution is based on the Proof of Work (PoW) consensus mechanism.
- Chapter 5 evaluates the impact of the proposed protocol in chapter 4 against malicious vehicles (i.e., vehicles spreading erroneous messages).
- Chapter 6 presents a secure and scalable framework for traffic event management based on Practical Byzantine Fault Tolerance (PBFT) consensus mechanism. In this protocol, a group of RSUs is dynamically formed based on the event appearance location to improve the road event validation.
- Chapter 7 presents a trust model that enhances the blockchain performance and isolates malicious RSUs. The goal is to ensure decentralized block creation while minimizing block validators' group size.
- Chapter 8 concludes this thesis and discusses the perspectives.

2 General Background : VANET & Blockchain Technology

Contents

2.1 Introduction

This chapter starts with the basics on Vehicular Ad hoc NETworks (VANET) architecture and goes through VANET characteristics, applications, and security

challenges. It also provides a general overview of blockchain concepts such as the consensus mechanism, blockchain properties, applications, and challenges.

2.2 VANET

Vehicular Ad hoc NETworks (VANET) is based on the Mobile Ad hoc NETwork (MANET), which is an infrastructure-free and self-configuring network of wireless mobile devices. VANET typical architecture is composed of three important actors : The Trusted Authority (TA), RSUs, and the vehicles as shown in Figure 2.1.

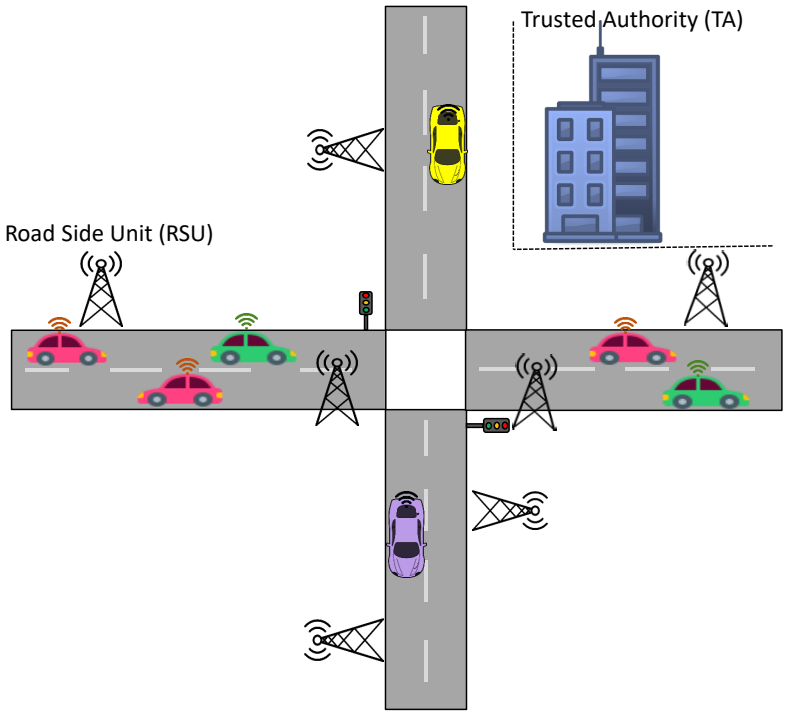

Figure 2.1: A typical VANET architecture

The TA is a trusted third party that controls and monitors the Vehicular Network (VN); it can revoke vehicles from the VN and help track misbehaving vehicles. The second element is the RSUs, which is, regularly, network infrastructures supporting vehicles with connectivity, computation, and storage resource. RSUs can act as base stations (e.g., WiFi, WiMAX) to be the intermediate layer between the TA and the vehicles. Finally, the vehicle, as the leading actor of VANET, is equipped with sensors that allow it to evaluate the traffic situation. It is also equipped with the Dedicated Short Range Communications (DSRC) module, which enables communication with the RSUs and other vehicles. In addition,

DSRC provides Wireless Access in Vehicular Environments (WAVE) capability using the IEEE 802.11p standard, providing medium access control (MAC) and physical (PHY) layers [20]. With this standard, a vehicle can communicate within a 1 km radius with transmission speeds ranging from 6 to 27 Mbps. DSRC is adopted in the USA standard; its equivalent is ITS-G5 in the European standard for Intelligent Transportation Systems (ITS).

In Vehicular Ad hoc NETworks (VANET), we can distinguish three types of vehicle-originated communications : Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I). Vehicle-to-Anything (V2X) extends vehicle communications to all entities, such as pedestrians. There are also communication scenarios such as Anything-to-Infrastructure (X2I).

TA, RSUs, and vehicles form a Cooperative Intelligent Transportation System (C-ITS) that aims to improve the safety and efficiency of the transportation sector.

2.2.1 General characteristics

VANET has specific characteristics that need to be considered when designing its applications and protocols.

- **Dynamic network**: vehicles are very dynamic and change their position and speed rapidly, making the network topology dynamic. This significant change in the network will impact communication performance. It also leads to network partitioning. Generally RSUs are relied upon for better connectivity.
- **Predictable mobility**: vehicle mobility is limited by road topology, traffic signs, and signals. Therefore, defining a mobility model is essential and has an impact on the whole system.
- **Large scale**: VANET is large in dense urban areas. This increases the communication load between the vehicles and also increase the requests load to the RSUs.
- **High computational ability** : vehicles are outfitted with modern sensors, GPS devices, and advanced antenna technology. They are also supplied with AI functions, allowing them to assess exact data regarding their environment (current position, speed, and direction) in real-time. Thus, they can predict immediate close danger.
- **No critical energy constraint** : although vehicles have limited resources, they are less resource-constrained than most MANET configurations. They

have the computation and storage power to run a wide range of applications. However, the supported applications should not be so resource-intensive that it compromises vehicles' primary task: sensing their environment and disseminating relevant information to improve road safety and efficiency.

- **Different QoS requirements** : in VANET, the QoS is crucial. There is no better if all services could be in real-time. However, some services are more time-critical than the rest. For instance, services related to road safety need a real-time response while other services are more flexible, such as traffic efficiency and driver comfort-related services.
- **Central registration** : vehicles as well as RSUs are registered with the TA and have unique identity (i.e., certificate).

The above specific characteristics make the context unique and exciting to investigate. Therefore, VANET applications based on their specialization must consider the above characteristics.

2.2.2 Applications

The advance in electronics and the current trend in wireless networks enable different deployments of Vehicular Network (VN). VN can be configured for various environments and to support multitudes of applications. These applications are meant to improve the safety and comfort of each entity in the transportation system (e.g., drivers, pedestrians, etc.). VANET can be classified into three broad categories: safety applications, efficiency applications, and comfort applications based on their essential objectives :

- **Safety applications:** are specialized in accident avoiding; they are based on emergency-related messages distribution to avoid accidents. These applications are very time-critical and generally represent inter-vehicular communications. They rely on CAM (Cooperative Awareness Messages), which gives the vehicle's current state (i.e., position, speed, direction, etc.) to predict an accident. A typical use case of such applications is advertising a hazardous event (e.g., sudden break) to approaching vehicles.
- **Efficiency applications:** this category of applications focuses on improving traffic efficiency and is less time-sensitive than security applications. There are many examples of such applications. To name a few, mobility applications: aim to optimize decision-making for drivers; environmental applications: provide real-time support for environment-friendly activities such as

eco-driving; travel plan optimization: reduce time, cost, and CO2 emissions; real-time traffic monitoring; and urban city surveillance.

This class of applications requires high availability and reliability. Its primary focus is traffic management and traveling services improvement.

- **Comfort applications:** this class of VANET applications concerns entertainment and comfort services; its main focus is to improve driver and passengers' comfort. Comfort applications provide accommodation services, restaurants, gas stations, tourist information, city and parking information, and sales announcements during a trip.

Proposed solutions in this thesis aim to make VANET applications more secure. In the coming section, we highlight the security requirement of these applications.

2.2.3 Security requirements

VANET applications are meant to provide all required services for a safer and more convenient transportation system. However, to secure these services, specific security requirements must be satisfied. In summary, these requirements are [21] :

- **Authentication :** the service provider should be authenticated and legitimate
- **Availability :** the service should be available and accessible
- **Flexibility and efficiency :** response delay must be minimized
- **Data integrity :** provided data must not have been altered, and its trustworthiness should be verifiable
- **Error detection :** detecting malfunctioning and attacks
- **Non-reputation :** service provider can't deny its actions
- **Confidentiality :** communication channel must be secure
- **Vehicle privacy must be guaranteed**

VANET is prone to security and privacy attacks that threaten human life and cause other social and economic disasters. These attacks are from authenticated entities (insiders), taking advantage of their knowledge of the network configuration to spread paddle, or from non-authenticated entities (outsiders),

seeking to penetrate the network and perform malicious actions. In addition, both vehicles and RSUs are subject to vulnerabilities. RSUs are widely spread with no protection and are prone to attacks that compromise their credibility. As results, RSUs can be malicious and disseminate erroneous information in the Vehicular Network (VN) [21]. In what follows, we briefly cite examples of attacks that might compromise the above-mentioned security requirements.

Attacks on integrity: attacks on integrity consist of falsifying exchanged data between the system entities (i.e., vehicles and RSUs), which disturbs road traffic safety. A typical example of these attacks is the *Bogus Information Attack*, wherein a vehicle/RSU spreads wrong messages to other entities of the network for its profit. For instance, a car A could send "There is a jam on the road R," aiming to free the road for itself [22]. Therefore, ensuring that traffic-related messages have not been altered is primordial to avoid disorders and accidents on the road. Another attack that affects data integrity in VANET is *Timing Attack*. This attack delays warning message dissemination [23].

Attacks on availability: vehicles and RSUs can perform *Spamming attacks*, i.e., sending spam messages in intention to increase traffic-messages transmission latency. Besides, they can be inefficient or unresponsive due to an overload of dummy messages, resulting from *DoS Attacks* [24]. Furthermore, the network can be subject to *Malware attacks*, where malicious software contaminates other nodes in the network and potentially turns them down. Moreover, nodes can reject participating in data dissemination throughout the system, causing loss of data; this is known as *Back Hole attack* [24].

Attacks on authentication : secure node authentication is vital for the security and the trustworthiness of the shared data in the VN; therefore, the system must avoid identity falsification. Various attacks can arise from authentication. A typical example is *Sybil Attack* which consists of sending numerous wrong messages using fake identities [25]. As a consequence, the attacker can misguide vehicles for its profit. Another attack that affects the authentication is the *Position Faking Attack*, in which the attacker can falsify its real position and takes that advantage to report wrong events [25]. Furthermore, when nodes do not have unique and tamper-proof identities, during messages exchange between two entities, the receiver can modify the message content before broadcasting it throughout the system as if it was the originator of the original message. This is known as *Node impersonation attack* [25].

Attacks on confidentiality: an attacker can perform the *Eavesdropping Attack*, seeking to sniff the communication between two nodes and to intercept and steal confidential information [25]. Therefore, it is essential to secure the communication channel.

In summary, there are vulnerabilities at both the Ad hoc layer (vehicles) and the RSUs layer. Therefore, securing VANET services is an essential step towards more secure transportation systems.

Therefore, this thesis focuses on proposing more robust traffic efficiency applications. Data integrity and availability are the primary goals of this work. We believe that blockchain technology has the potential to meet these requirements if it is well adapted to VANET.

2.3 Blockchain Technology

As the name suggests, blockchain refers to a chain of blocks linked by a cryptographic hash. This data organization concept was first introduced in 1990 by Haber and Stornetta in their paper entitled "How to Time-Stamp a Digital Document" [26]. However, blockchain technology was put prominent by Nakamoto, at the end of 2008, as the underlying technology of Bitcoin [18]: the first blockchain-based cryptocurrency. Since then, this technology has promised to revolutionize the financial market. Four years later, Ethereum [27] took its place as the first blockchain platform supporting no economic assets through Smart contracts (which can define any type of information). That was a big boost for blockchain technology adaptation and opened horizons to new applications. Over the past decade, blockchain technology has demonstrated its ability to reconfigure the economy and has promised to revolutionize the industry, politics, and legal systems [28]. The blockchain market size was estimated at USD 3.67 billion in 2020 [29]. On the academic side, institutions such as Stanford University (USA) and Beijing University of Aeronautics (China) have embarked on a race to develop blockchain technology and its applications towards maturity [30].

2.3.1 Basic structure of blockchain

From a data structure perspective, a blockchain is essentially an ordered history of transactions (e.g., atomic data) organized in blocks. As illustrated in Figure 2.2, each block encompasses a set of transactions, and each block is linked to its predecessor by a cryptographic pointer (hash). Transactions within a block

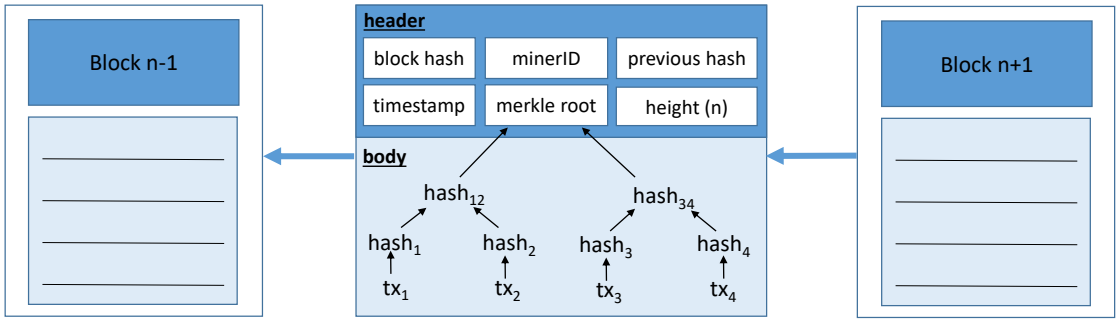

Figure 2.2: Typical structure of a Blockchain

are compressed into a single hash called the *merkle root*, which is calculated by pairwise hashing until only one hash remains. This hash is later used with the rest of the block header fields (e.g., timestamp, previous hash, etc.) to compute the block hash. Subsequently, the resulting hash will be included in the next block as *previous hash* to form a cryptographical link between the blocks. To summarize, a blockchain, in its simple presentation, is a distributed ledger that is immutable. This property is crucial for transparency and public verifiability of the ledger. Nevertheless, none of these properties can be guaranteed in a trustless environment without a secure consensus mechanism.

2.3.2 Consensus

The consensus is the primary component dictating the security and performance of a blockchain system. Its goal is to ensure a synchronized ledger between blockchain nodes in the presence of a bounded number of Byzantine faults (i.e., malicious/arbitrary malfunctioning). Formally, a resilient consensus protocol must meet the following three properties [31] : 1. Consistency: a final agreement is reached; 2. Validity/Integrity: all correct nodes decide the same value; 3. Termination: eventually, each consensus node decides some value. The aim behind a consensus is to solve the General Byzantine Problem which was formalized by Lamport et al. [32] in 1982. The authors concluded: this is a difficult problem, and its solution seems to inherently require many message exchanges. In fact, a solution to this problem exists only under some hypothesis [33]. Below, the general assumptions that can be made based on the communication reliability.

- **Synchrony:** communication delays and process speeds are bounded, and the upper bounds are known.
- **Partial synchrony (Weak synchrony):** there exist unknown time slots during which the system is synchronous.

- **Asynchronous:** communication delays can be infinite. the same for the process delays.

Although the last assumption is more realistic, Fischer et al. proved that consensus could not be achieved in an asynchronous network with the presence of one fault process: it is the FLP impossibility result [33]. Therefore, partial synchrony was introduced by Dwork et al. [34] to deviate from the FLP impossibility.

Traditional consensus algorithms, such as Paxos [35], do not support arbitrary faults (Byzantine faults); only benign (crash) faults are allowed. However, in a trustless network, consensus participants may intentionally broadcast wrong information to attack the system. Therefore, in the last 20 years, consensus algorithms supporting Byzantine faults have been the focus of the distributed systems' community [19, 36, 37].

Particularly in the blockchain context, several consensus protocols have been proposed. They can be classified into two categories: voting-based consensus mechanisms and proof-based consensus mechanisms.

The proof-based consensus protocols are based on a competition between consensus participants. Examples of such protocols are the Proof of Work (PoW), Proof of Stake (PoS), Proof of Elapsed Time (PoET), and Proof of Authority (PoA).

- **Proof of Work (PoW)** also called "The Nakamoto consensus," has been widely adopted since its prominence as the underlying consensus mechanism of Bitcoin. PoW consists of a cryptographic puzzle-solving, whereby miners (i.e., consensus participants) race to propose the next block of the blockchain. More explicitly, miners allocate their dedicated CPUs to find a block with a hash verifying a specific pattern. Then, the puzzle winner proposes the next block, which contains the previous block's hash. Next, the winner appends the block to its local copy of the blockchain and then forwards the block to its peers. Finally, the block is easily verified by the network by performing a hash comparison. PoW-solving generally requires an incentive system to encourage miners in maintaining the blockchain.

  Forks:

  Sometimes, a block can be created by different miners at the same time so that these blocks are valid and have the same parent block. However, this eventual collision engenders inconsistency in the blockchain. Because when these two blocks are propagated in the network, both blocks are appended

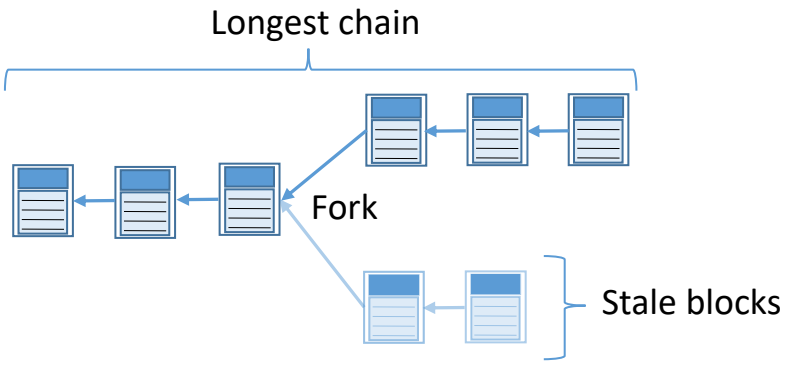

Figure 2.3: Blockchain fork

to the blockchain. In this particular situation, the blockchain is in a fork, as illustrated in Figure 2.3. Note that the same transaction can be in both blocks; therefore, to ensure consistency in the ledger, only one block should be considered. The other blocks will be dropped, and their transactions will be saved back in the miner's *memepool* (stores all unconfirmed transactions) for the next blocks.

In Bitcoin, for example, forks are resolved following the *longest chain rule*: the main chain is the one that has the most significant PoW effort invested in it [18]. Blocks not belonging to the longest chain are called "stale blocks." Although these blocks are valid, they are wasted and will be dropped from the blockchain. Moreover, from the security standpoint, stale blocks increase the advantage of the adversary in the blockchain network [38]. Fork resolving takes time; i.e., transactions cannot be considered immediately persistent in a PoW-based blockchain, no immediate transaction confirmation (finality). As a result, this directly affects the latency of the PoW protocol. PoW-based consensus protocol requires a transaction to be sufficiently old before being spendable/exploitable. For instance, in Bitcoin, it's suggested to wait for 6 blocks before considering a consistent block. i.e., a transaction has to wait an hour to be spendable.

When implementing PoW, balancing the PoW difficulty (i.e., the average time for a new block to appear) and fork appearance is essential. Indeed, the easier the PoW puzzle, the higher the fork probability. This is why, in Bitcoin, the PoW difficulty is configured so that it takes 10 minutes on average to mine a block [1].

Regarding security, PoW was initially supposed to be secure as long the adversary does not hold more than 50% of the system's total computational

[1] Bitcoin wiki: https://en.bitcoin.it/wiki/Mining

power [18]. However, years later, it was shown by Eyal and Sirer [39] that even if the adversary has only 25% of the computing power, bitcoin mining becomes vulnerable.

- **Proof of Stake (PoS)** emerged as a solution to the PoW's high computational load cost. Instead of cryptographical puzzle-solving, PoS uses stake (e.g., a share deposit) so as block mining chances are proportional to the miner's stake. The reliability of PoS lay on the miner's interest to behave correctly to protect its stake and eventually gain the mining reward. In the opposite case, its deposit will be lost. PoS is resource-friendly compared to PoW, which involves high computational and energy costs. However, PoS is not completely mature; among its current challenges, the miner election. For example, the mining criteria cannot be based on the highest deposit. Otherwise, that will lead to a monopoly of high-stake holders and therefore affects the fairness and decentralization of the blockchain. As a matter of fact, the block miner must be random and unpredictable to avoid malicious miners to bias the election process. Therefore, techniques such as coin-age are relied on to randomly elect a miner among stakeholders [40, 41]. However, generally, these techniques require a synchronous network [42].

- **Proof of Elapsed Time (PoET)** replaces the PoW-puzzle solving by a random waiting time. The aim is to impose miners a random waiting delay before proposing a new block. To do so, an especial environment called Trusted Execution Environment (TEE) that enables isolated and secure execution of trusted applications is relied on. Moreover, sophisticated software such as Intel Software Guard Extensions (XGS) is required to generate the random waiting period for miners as well as a proof of their waiting time. Basically, PoET works in two phases: each validator is assigned a random waiting period by the network (trusted code source) [43]. The validator with the shortest time wins and gets a certification to append a new block to the chain. PoET is optimal as for communication, energy costs. Nevertheless, the XGS provider (Intel) should be trusted [44].

- **Proof of Authority (PoA)** relies on identified and trusted validators called authorities or *sealers*, whereas the majority of them are honest. Authorities are formed based on their reputation. PoA is supposed to support $t = n/2$ Byzantine faults from n fixed authorities. However, this complies with the impossibility to solve consensus when $t \geq n/3$ [45]. Therefore, PoA requires synchrony to be secure [46].

On the other hand, we have the voting-based consensus mechanism or Byzantine Fault Tolerance (BFT) consensus family. Such protocols require authenticated

participants. And work on rounds of communications to reach an agreement between this latter. Examples of such consensus mechanisms are Practical Byzantine Fault Tolerance (PBFT) [45], Ripple consensus [47] and Stellar Consensus Protocol (SCP) [48].

- **Practical Byzantine Fault Tolerance (PBFT)** was the adopted consensus protocol by IBM in its famous blockchain platform Hyperledger Fabric. PBFT is similar to Paxos [35] with an extra round of communication, enabling it to support Byzantine faults. Effectively, PBFT has been proven to be safe under an upper bound of $t \leq n/3$ Byzantine faults of a n consensus participants, which is shown to be optimal [45]. PBFT works by rounds, such as in each round, nodes perform three rounds of communication ($Pre-prepare$, $Prepare$, and $Commit$) to reach a consensus on a given value. Initially, a node is elected as leader (primary) to lead the voting process, whereas other replicas act as backups as indicated in Figure 2.4. The consensus leader could fail. In that case, the backups nodes perform a *change view*, and the current leader will failover with one of the backups. The *change view* process is mandatory to ensure the progress of the system. PBFT guarantees both liveliness and security under these three assumptions [19]:

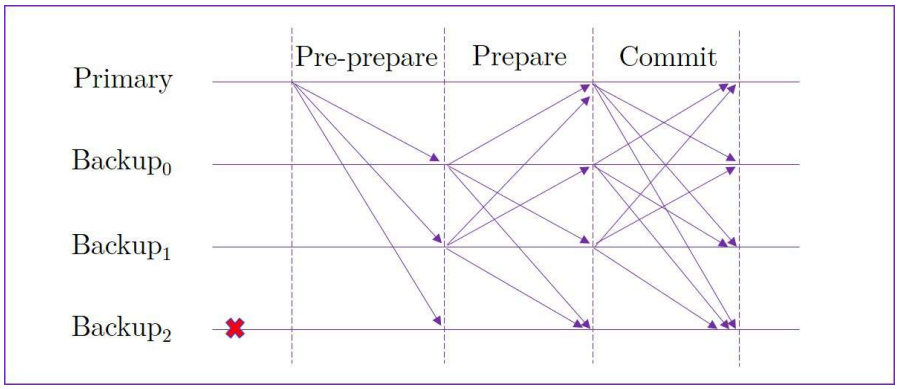

Figure 2.4: Practical Byzantine Fault Tolerance (PBFT) three phases of communication [19].

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

In PBFT, consensus participants identities must be known to ensure exchanged messages authenticity. 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 ensures that the leader sends the same $prepare$ message to the correct backups. After receiving $2t+1$ $prepare$ messages matching the $pre-prepare$ message received earlier from different nodes, a node will broadcast a $commit$ message and moves to the $Commit$ phase. Like the $Prepare$ phase, after receiving $2t+1$ commits corresponding to the $pre-prepare$ message, all valid nodes will add the block to their local copy of the blockchain. See Figure 2.4.

Note that PBFT requires $O(n^2)$ messages exchange in the normal scenario and $O(n^3)$ for the *change view*, where n is the size of the consensus group. This communication cost affects the performance of PBFT.

- **Ripple** is a blockchain platform that uses a voting-based consensus. Although Ripple's consensus protocol is voting-based, it's different from PBFT. In the former, a transaction has to be approved by 80% of the network. Therefore, each consensus node needs to maintain a Unique Node List (UNL), such as this list's intersection with any other list from another consensus node is at least 1/5 of the total nodes [47]. From security perspective, Ripple's consensus mechanism does not support more than $(n-1)/5$ Byzantine faults among n consensus nodes, which are far from optimal.

- **Stellar Consensus Protocol (SCP)** is another different voting-based consensus algorithm. It uses a special hierarchical connection between the consensus nodes to form groups (quorum slices). These quorum slices are formed based on the trust between the consensus nodes [48]. Mazieres [48] claimed that Stellar achieves optimal resilience against Byzantine nodes (supports $n/3$ Byzantine faults). However, Stellar relies on individual trust decisions between the consensus nodes that orchestrate the consensus on their behalf.

2.3.3 Types of blockchains

Existing blockchains can be classified into two broad categories based on writing access: permissionless and permissioned blockchains. The former is suitable for public networks, as no authentication is required for consensus nodes. Bitcoin and Ethereum are examples of these open/permissionless blockchains since miners can join and leave the network at any time with no restrictions. However, unlike open blockchains, permissioned blockchains require authorization for consensus participation. Typical examples of such platforms are Hyperledger Fabric [49],

Quorum [50], and Corda [51]. This classification could be detailed further into public blockchains, consortium blockchains, and private blockchains as shown in Figure 2.5

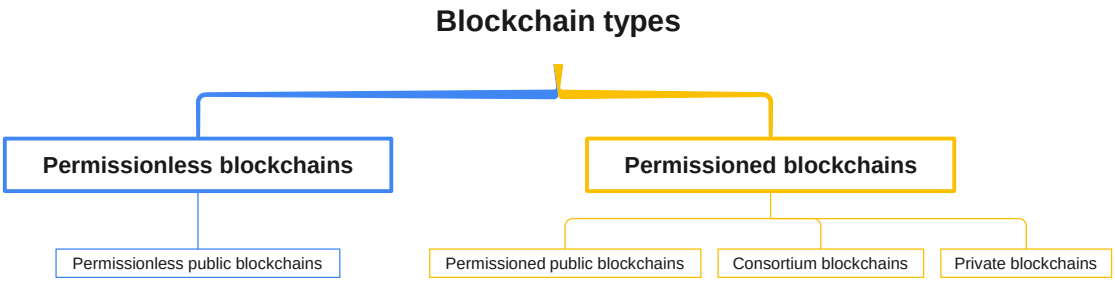

Figure 2.5: Types of blockchains

Public permissioned blockchains: each node has read access to the blockchain and has the freedom to join/leave the network. However, to become a miner (writer), specific conditions should be fulfilled. For example, reaching certain credibility or owning a particular stake over the network.

Public perimissioneless blockchains: no permission is required to become a consensus node. The read access is granted to everybody, and each consensus node has the total access to the blockchain.

The shared point between public permissioned and public pemissioneless blockchains is that each node in the network has read access to the blockchain. In addition, each node can potentially grant the writing access by fulfilling certain conditions. Therefore, they are highly decentralized.

Private blockchains: an authority can unilaterally change the blockchain rules. This type of blockchains can be useful in case of a large organization treating with various partners as subcontractors. The organization can unilaterally validate its subcontractors services or decide to end the contract. For example, private blockchains are highly adopted in healthcare systems because of critical data access and management policies that may involve the government [52].

Consortium blockchains: entities sharing interests define the blockchain rules. Thus, the blockchain is maintained between this group, wherein each participant represents its shares and contributes to the system's progress–any update on the blockchain rules need approval from the majority.

2.3.4 Applications

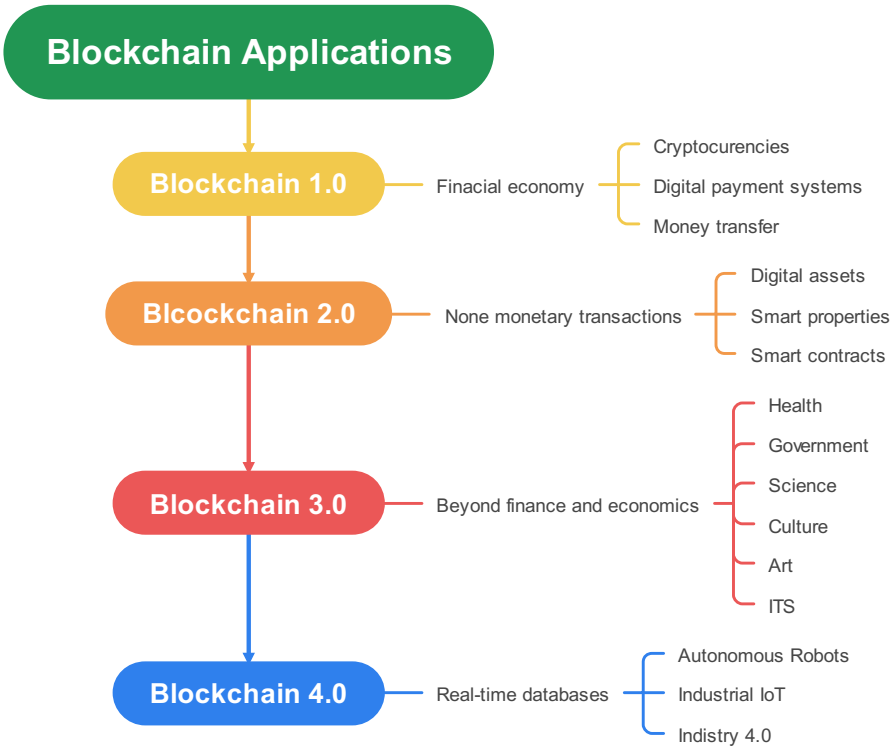

Figure 2.6: Types of blockchains

Over the past decade, blockchain technology has demonstrated its ability to reconfigure the economy, politics, and legal systems. Based on its application domains, the blockchain technology evolution is classified as follows: Blockchain 1.0, 2.0, 3.0, and 4.0 [53]. Figure 2.6 illustrates blockchain applications in various areas. Blockchain 1.0 concerns the financial economy, i.e., cryptocurrencies and related applications such as digital payment systems and money transfer. Blockchain 2.0 is about contracts, where more complex data than monetary transaction are involved, i.e., titles, digital assets, smart properties, and smart contracts. Ethereum leads this generation of blockchain with the famous concept of smart contracts (i.e., digital agreements). Blockchain 3.0 covers applications beyond finance and economics, such as health, government, science, culture, and art. Although this generation is still preparing for mass adoption, we have already talked about Blockchain 4.0, which processes real-time applications-based blockchains. This gives an insight into the potential of blockchain to divert sectors of activities, such as in industry 4.0.

2.3.5 Properties

- **Decentralization :** blockchain technology allows transactions to be validated without relying on a central authority, thanks to the decentralization property. This latter enables each node in the blockchain network to provide the system's state, independently. As a result, services remain available to users even with the presence of attackers and crashes. It also mitigates the risk of single-point failure by limiting requests to a centralized data center. Moreover, the decentralization reduces the performance bottleneck at the central agency level [54].
- **Transparency/Public verifiability :** the blockchain state is independently verifiable by each node of the network (maintaining the blockchain). As such, the database's state is updated without the need to communicate or trust any central authority.
- **Immutability/Tamper-resistant :** blockchain is a chain of blocks linked by a cryptographic hash from the last block to the first block of the chain (*genesis block*). Thus, any modification of a block's content breaks the chain and is easily detected, hence its immutability.
- **Redundancy :** the blockchain is replicated on each node of the network, i.e., each miner (consensus participant) stores and maintains a local copy of the blockchain. That increases the fault tolerance and, therefore, the robustness of the system.
- **Integrity :** by transparency and immutability, the blockchain also guarantees data integrity as long as a secure consensus mechanism powers it. All/part of the network approves all blocks through a transparent consensus mechanism.
- **Non-repudiation :** transactions are signed, relying on digital signatures, before being added into the blockchain and becoming persistent. As a result, no entity can deny its activities towards the system.

2.3.6 Challenges

With these properties, blockchain technology has almost found applications in all fields. However, despite having numerous advantages, most of the existing blockchain-based protocols face scalability limitations: high transaction confirmation time, storage, and computation overhead, which hinder the complete adoption of this technology for many use cases. Therefore, much research has been

conducted in that regard; and various solutions have been proposed. Some have focused on introducing more optimized consensus algorithms [55, 56]. Others proposed to *shard* the network [57, 58], i.e., partitioning consensus participants into multiple subgroups working in parallel. Differently, processing transactions outside the main chain have also been suggested; it is known as the off-chain approaches [59]. Furthermore, DAG (Directed Acyclic Graphs) [60], another form of a distributed ledger, was introduced as an alternative to the traditional linked list of blocks structure, aiming to process more transactions in parallel. Finally, federated approaches, also called committee-based solutions, have been proposed. They consist mainly of minimizing the consensus group size. See Figure 2.7.

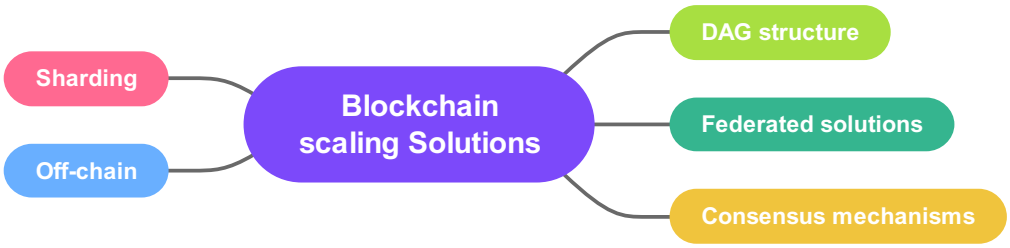

Figure 2.7: Blockchain scaling solutions

The primary distinction of committee-based schemes is on how committees are formed. For instance, Byzcoin [61] uses a fixed sliding window size of recent proof-of-work block miners to form a committee dynamically. In Tendermint [62], nodes join a committee by positioning a bond-deposit. And Algorand [63] relies on a random function to form a committee. In these approaches, committee forming requires proof of membership, which is satisfied by performing PoW or PoS. However, in our proposal, there is no need for proof-of-membership since consensus participants are authenticated in advance. More details about the protocol will be provided in subsequent sections.

In addition, committee-based approaches are usually built on a voting-based consensus algorithm. Byzantine Fault Tolerance (BFT) consensus algorithms are widely adopted for that purpose, as they offer low latency and high consistency [64]. Practical Byzantine Fault Tolerance (PBFT) is a typical example of BFT algorithms. Hyperledger Fabric [49] was the first platform to implement PBFT in the blockchain framework; since then, many other blockchain platforms have been built on PBFT.

In this work, committee-based approaches were relied upon to enhance our adapted blockchain for road traffic data management.

2.4 Conclusion

This chapter gives general background for a better understanding of the upcoming chapters; it is twofold: Vehicular Ad hoc NETworks (VANET) and blockchain technology. The first part of this chapter is dedicated to VANET. VANET architecture is introduced, and its characteristics, application requirements, and security challenges are presented. The last part introduces blockchain technology and gives general knowledge about its properties, applications, and challenges. Throughout this chapter, we briefly related our contribution to the background knowledge. The next chapter will give a thorough review on blockchain technology adoption in Vehicular Network (VN).

3

Vehicular Network Architectures : Review

Contents

3.1 Introduction

This chapter reviews blockchain technology adoption in Vehicular Network (VN); it gives a taxonomy of Vehicular Ad hoc NETworks (VANET) architecture evolution up to blockchain technology integration. The taxonomy classifies existing VANET architectures into three broad categories as shown in Figure 3.1: centralized, distributed, and decentralized schemes. Adapting blockchain into VN context is none-trivial, and several attempts have been made. This chapter discusses each of these architectures and highlights its limitations toward secure traffic data management. Moreover, blockchain adoption in the VN, as well as the gaps in the state-of-art, are discussed.

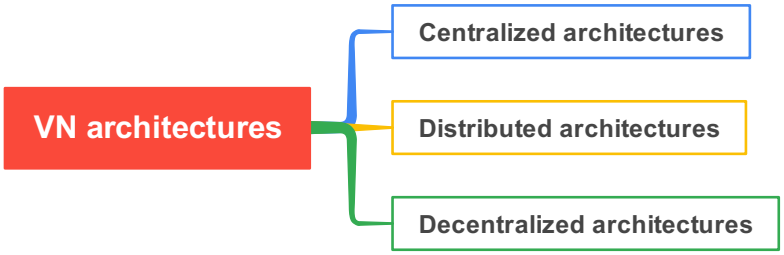

Figure 3.1: VN architectures

3.2 Centralized Architectures

Autonomous vehicles are equipped with numerous sensors, cameras, and LiDAR sensors, which generate a massive amount of data for each car. The rate of vehicular data generation is increasing dramatically. For example, Google's autonomous car generates a data stream between 1.4 TB/hr and 19 TB/hr [10, 65]. This huge amount of data is crucial for improving road traffic safety and efficiency. An autonomous vehicle needs to analyze and process a huge amount of data sensed through its embedded sensors or handed by surrounding vehicles to make safe decisions.

Over the past decade, Cloud Computing (CC) has emerged as a promising solution for large-scale data storage and computation [66, 67]. In particular, VANET architectures depended on cloud computing to cope with vehicular data [68–71]. The primary benefit of coupling the cloud with VANET is to reduce the computation and storage load on vehicles by offloading heavy tasks to the cloud. In addition, CC provides unlimited computing and storage services through remote

servers. These servers function as a central datacenter that gives anywhere and anytime access to authorized vehicles.

In addition to computing and storage, the cloud offers other services such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). From an economic standpoint, these services allow a pay-as-you-go model [72, 73], making them more attractive for automakers. For example, Toyota relies on Microsoft Azure HDInsight to manage and process the astronomical amount of data generated by its vehicles [10][p. 16]. As a result, Toyota can offer services to its cars on-demand, directly via the internet.

This concept of offloading tasks to the cloud has been introduced with the emergence of the Mobile Cloud Computing (MCM) paradigm. The aim is to mitigate mobile devices performance and resource obstacles (e.g., battery life, storage, and bandwidth), environment (e.g., heterogeneity, scalability, availability), and security (e.g., reliability and privacy) [10, 74]. Similarly, this idea has been driven into VN to support vehicles with unlimited storage and computation resource through CC. This concept is known as Vehicles using the Cloud (VuC); it improves Intelligent Transportation Systems (ITS) applications such as real-time traffic prediction and enables various web services [71, 75].

3.2.1 Vehicles using Clouds (VuC)

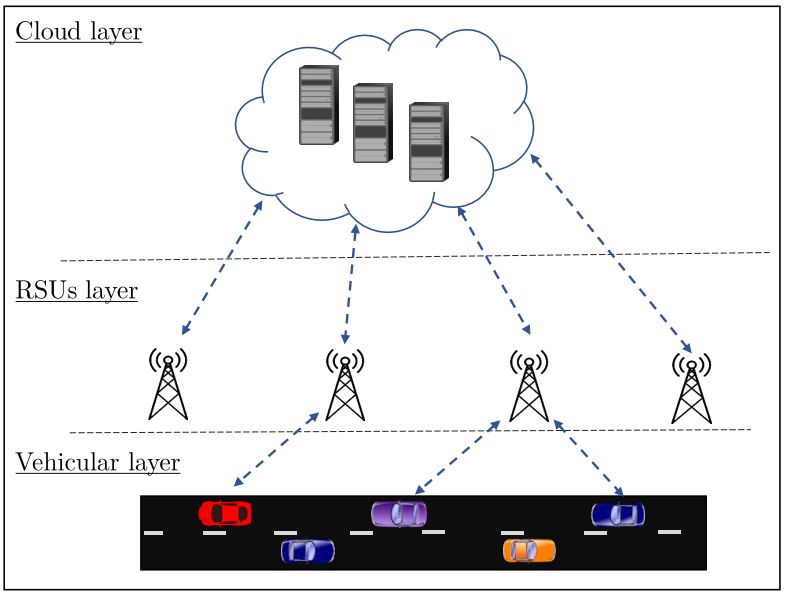

Figure 3.2: Vehicles using Cloud (VuC)

As shown in Figure 3.2, the VuC architecture is defined so that vehicles can interact with the cloud, directly or through RSUs serving as gateways. Thus, vehicles contribute with valuable data that will be processed and analyzed by the cloud for decision making. After processing the vehicular data, the cloud provides real-time traffic information and infotainment services to authorized vehicles. In addition to big data analytics and traffic messages advertising, many practical applications have relied upon the VuC concept. To cite just a few, Hyundai [76, 77] relies on a central structure to monitor configuration and performance of its vehicles; the goal is to improve its customers' experience and service quality. Note that in such applications, vehicle privacy is a serious concern. Furthermore, Hussain et al. [78] introduced the VWaaS (Vehicles Witness as a Service) to monitor and report accidents and other dangerous events in road traffic. The proposed application enables vehicles to upload accident images to the cloud so that these latter can be used as forensic evidence to law enforcement and insurance agencies. Moreover, Bitam and Mellouk [79] proposed a traffic data dissemination protocol relying on the cloud. This approach aims to reduce the burden on vehicles while processing traffic information transmitted from other cars.

VuC design is attractive because the cloud provides valuable services to vehicles and reduces the storage and computation load on them. However, VuC inherent convention clouds security and performance limitations which prevent it from meeting VANET applications requirements. These limitations are mainly due to the distance between the cloud and the vehicles, which causes a lack of mobility support, geo-distribution, location awareness, low latency, and security attacks because of long-distance transmission.

3.2.2 Vehicular Cloud Computing (VCC)

Vehicular Cloud Computing (VCC) has emerged as a new paradigm that leverages underutilized vehicle resources to form a cloud. Instead of relying on traditional cloud providers such as Amazon or Google, as is the case for VuC, in VCC vehicles create a shared cloud. For example, a fleet of dynamic cars (e.g., cars with similar mobility patterns) or stationary cars (e.g., cars in shopping malls or parking lots) can leverage their unused storage and computation resources to cooperate by forming a cloud system as illustrated in Figure 3.3. The formed VCC enables various valuable such as Storage-as-a-Service (StaaS), Computation-as-a-Service (CaaS), Network-as-a-Service (NaaS), and Sensing-as-a-Service (SaaS) [80]. Thus, vehicles can request the vehicular cloud to access traffic information and other services. VCC has many advantages over conventional Cloud Computing (CC) [81]. One of the key properties of VCC over VuC is its ability to reduce vehicle requests processing latency.

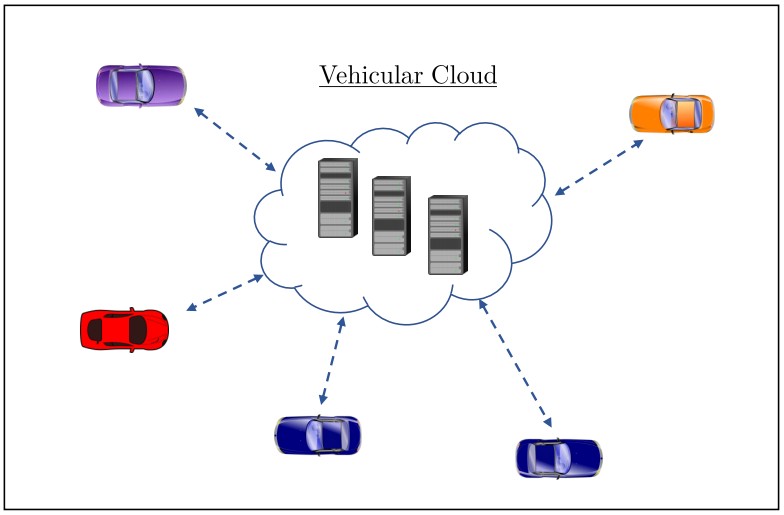

Figure 3.3: Vehicular Cloud Computing (VCC)

VCC is supposed to offer valuable services such as traffic safety management, road traffic information dissemination, infotainment, and disaster management [71, 81]. For example, a group of vehicles can form a cloud wherein they exchange and maintain traffic information without uploading the data to a remote cloud. As a consequence, that minimizes long-distance data transmission; however, VCC raises critical security threads [82] that compromise data integrity, services availability, and confidentiality. For example, a malicious vehicle, as authenticated node in the VCC, can falsify data, disclose information, or intentionally stop its services, leading to data volatility. Furthermore, with a dynamic fleet of vehicles, partition and packet loss makes the VCC unpractical. For these reasons, the VCC is not suitable for road-traffic data management.

3.2.3 Hybrid Vehicular Cloud (HVC)

Cloud integration in the vehicular network has taken a step further, with Hybrid Vehicular Cloud (HVC) architecture which combines VCC and VuC [70]. The goal of this paradigm is to leverage both approaches simultaneously as shows Figure 3.4. HVC is useful when processed data can either be uploaded to a third-party cloud or kept locally in the vehicular cloud. In this type of design, the vehicle request is first submitted to the vehicular cloud before requesting the third-party cloud service. In this case, the benefits of both the VCC and VuC are found in HVC.

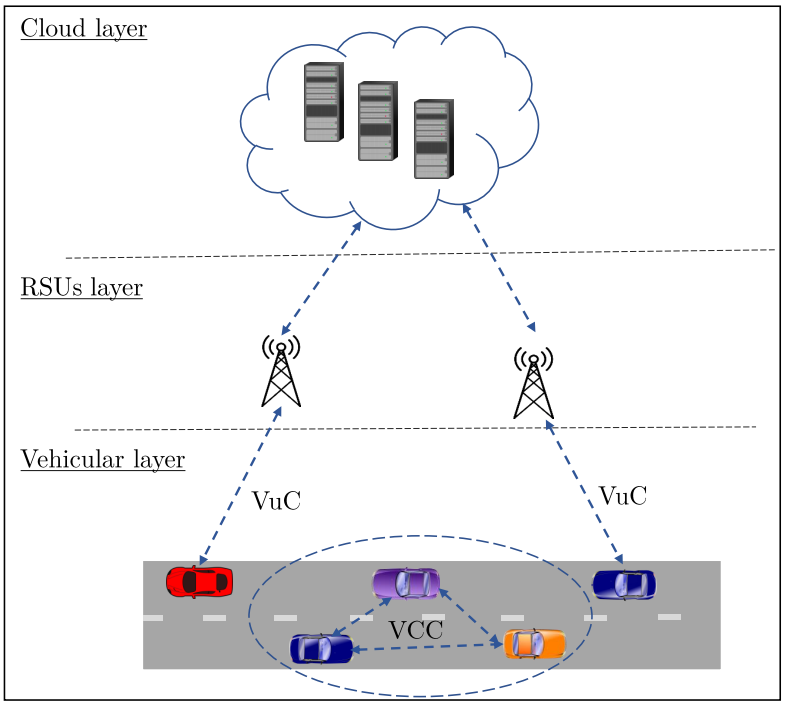

Figure 3.4: Hybrid Vehicular Cloud

3.2.4 Limitations of VANET-based Cloud Computing (V-CC)

V-CC is a rapidly growing concept. However, it is still in an immature stage, with many security and performance challenges that prevent meeting VANET applications requirements; especially, for road traffic data management, which requires high integrity and availability.

Security issues : cloud-enabled vehicular networks have some security and privacy issues. Due to the sensitivity of shared data between vehicles, there is a growing concern about vehicle privacy. Also, due to the high dynamic in cloud participants, in the case of VCC and HVC, authentication causes permanent weakness. A non-authenticated or malicious vehicle intrusion could cause loss of time and money, or even death. Therefore, much research has been conducted to solve the problems of intrusion, authentication, and vehicle privacy violation [11, 12, 83, 84]. However, there are still security and privacy issues in vehicular cloud computing [71][p 17] that affect data integrity. For example, in VCC, vehicles must rely on information from other vehicles or the cloud for navigation. Nevertheless, any distortion of the information can cause damage. That is likely to happen in the presence of malicious vehicles. Also, due to the lack of location awareness, it is

challenging to evaluate traffic information.

Performance issues : the increase in autonomous vehicles will imply excessive requests for traffic services. That will lead to network congestion and cause significant delays in accessing cloud services. Furthermore, the distance between the cloud and the vehicles will induce consequent data transmission delay. Therefore, V-CC can not satisfy the required low-delay response.

Moreover, excessive bandwidth and storage costs are implied from data uploading to the cloud, which increases network access devices' energy consumption. As a result, it becomes challenging for the cloud to keep up with the communication and computational demands with the continuous growth of vehicular data. For these reasons, V-CC architectures failed to meet VANET applications' QoS requirements, especially when it comes to traffic data management and sharing [10].

3.3 Distributed Architectures

Mobile Edge Computing (MEC), or similar paradigms such as Fog Computing and Cloudlets, are promising solutions to the above V-CC performance problems [9, 14, 85]. The key idea behind MEC is to bring the cloud closer to users. The goal is to minimize data offloading costs (e.g., bandwidth and energy) and reduce latency. For that end, cloud resources (i.e., computation and energy) must be split and distributed to edge servers located at the network's edge. Hence, the distribution property of the MEC-based architecture.

Over the past decade, MEC has attracted much attention in academia [86–89]. The MEC paradigm is well accepted by the research community. It is also considered as the practical solution to deal with the exorbitant mass of data generated by IoT devices. And for the specific, MEC has been introduced in the vehicular network as Vehicular Edge Computing (VEC), a revolution of the V-CC.

3.3.1 Vehicular Edge Cloud (VEC)

VEC enables MEC in the vehicular network, whereby cloud computing and storage services are brought closer to vehicles. That improves the QoS of the cloud by reducing the load on the network infrastructure and the bandwidth cost. For this reason, VEC holds promise for addressing the exceptional growth in vehicle demands and application requirements. The main property of VEC is its ability to reduce cloud resources accessing delay. In addition, VEC is, by design, distributed in opposite to V-CC, which is centralized [90]. Moreover, VEC enables data lo-

cation awareness, thanks to the proximity of the edge nodes. As a consequence, this makes it more efficient for road traffic data assessment compared to V-CC architectures.

On the other hand, VEC also helps balance data offloading and vehicle requests processing to mitigate potential network congestion. Although vehicles have high sensing capacity and moderate computation and storage resources, they still cannot effectively manage traffic data. Therefore, resource-intensive tasks must be uploaded to the cloud, whereby the result can be requested within a minimum delay to meet VANET applications specifics. In VEC, close edge nodes handle vehicle requests without needing the central cloud, thanks to optimized caching politics. As a result, the final delay is minimized, as well as the bandwidth and energy costs.

3.3.2 Architecture

As shown in Figure 3.5, a typical VEC architecture is composed of three layers: the vehicular layer, the Edge servers layer, and the cloud layer.

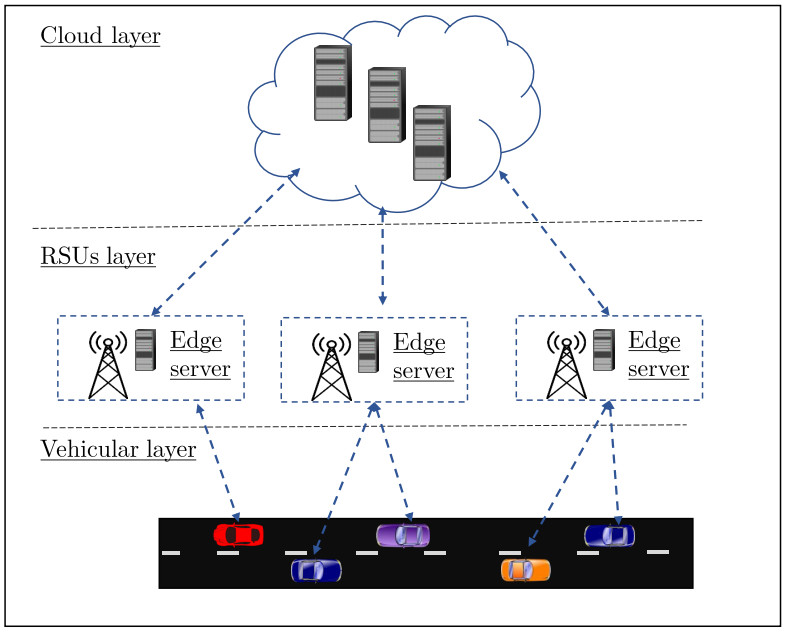

Figure 3.5: A typical VEC architecture

Vehicular layer : vehicles can process, analyze and store the necessary information perceived by its embedded sensors. However, they have limited computation and storage resources (cf.2.2.1); therefore, the need to offload ressource-

incentive tasks to more appropriate nodes (i.e., edge servers). The offloading process is through V2I communications.

Edge servers layer : Edge servers are often located at the RSUs layer. They support RSUs with enough computation and storage resources to process the vehicular data. In addition, if necessary, RSUs may forward tasks to the central cloud to access cloud services. Thus, the RSUs as well the conventional cloud work together to make the transportation system more convenient.

Cloud layer : The cloud layer comprises remote cloud servers operating as a central cloud with unlimited resources (i.e., computation and storage). Thus, it can cover a larger area than the edge servers. Finally, the cloud coexists with the edge servers to effectively reduce the load on them. For example, if cashed-out data at the edge is no longer relevant, it can be offloaded and transferred to the central cloud. In addition, the cloud can provide centralized control for optimal decisions.

3.3.3 Vehicular data management in VEC

An astronomical amount of traffic data is expected from vehicles. This information must be effectively analyzed, validated, securely stored, and disseminated. Attempts have been made in that regard while relying on the VEC architecture. For example, in a recent work Lai et al. [91] presented a VEC-based framework for an efficient and responsive vehicle requests processing. Furthermore, Hagenauer et al. [92] introduced micro clouds as edge servers that process and analyze the data before transferring it to a remote datacenter. Lai et al. [93] pushed the above-cited work further with a two-level threshold approach to dynamically optimize data transmission volume. Finally, Darwish and Bakar [94] proposed a combined massive data analysis with intelligent computing to enable real-time data processing in vehicles using fog computing.

VEC optimizes the overall QoS of VANET applications by minimizing bandwidth, energy, and cloud services accessing delay [85] . In recent years, many contributions have been made regarding VEC [95–100]. Ongoing research projects are investigating the inclusion of new technologies such as Software Defined Networking (SDN) [101–103] and Artificial Intelligence (AI) [104–106] seeking to further optimize VEC capacity. For example, given the limited resources of edge nodes, AI may be practical for optimizing resource caching.

3.3.4 Security challenges of VEC

Despite a low cloud service accessing delay, VEC comes along with security issues. Because of the high distribution of the edge nodes, they are difficult to protect. Consequently, edge servers might be malicious and thus compromise trust, confidentiality, data integrity, and transparency [85, 107] of the VEC system.

Trust and confidentiality : although vehicles, through their embedded sensors, provide relevant data to improve the transportation system, these data must be validated. With the explosive increase in the number of vehicles, those with malicious or faulty behavior overwhelm the safety and feasibility of the Vehicular Network (VN). Vehicle reputation management and authentication are the key measures to effectively assess the reliability of vehicles [85, 108]. Through a good reputation model, benign cars can build a strong credibility, and attacking ones can be detected and isolated. Therefore, an effective reputation management system will prevent potential attacks and thus increase the reliability and performance of the system. Researchers are working on designing an optimized and effective vehicle reputation management system [109–111]. These works focus on how to effectively develop an efficient reward/punishment model to update vehicle credibility. Less attention was given to potential vulnerability at the RSUs layer. Nevertheless, RSUs which are supplied with edge nodes are prone to attacks and errors (cf. 2.2.3) and therefore are not trusted.

Data integrity and transparency : RSUs are deployed in public places without any protection, making them prone to attacks and therefore not entirely reliable [85]. That said, to maintain data integrity between such a distributed and trustless system, data must be verified by a sufficient number of non-malicious RSUs [14]. Furthermore, ensuring transparency between these RSUs in an untrusted network is a major challenge [9, 14].

To sum up, VEC drastically reduces cloud resources accessing delay, which is a consequential added value for road-traffic services. However, it raises along the way new security challenges, such as the difficulty (i.e., expensive to supervise and maintain) to protect the edge servers located at the RSUs layer. To address these security issues, it is indispensable to ensure reliable and transparent communication between the edge servers.

We believe that blockchain technology, with its capacity to ensure data integrity and transparency in an untrusted network, is a promising solution to overcome VEC security limitations.

3.4 Decentralized Architectures

Blockchain technology properties, such as decentralization, tamper resistance, redundancy, and self-healing, can help achieve crucial security goals (cf.2.3.5). Therefore, its integration into Vehicular Network (VN) has gained considerable interest, lately [17, 112, 113].

3.4.1 Blockchain in Vehicular Network (VN) : Motivation

VN is ever vulnerable to attacks due to the high heterogeneity at different layers of the network and the intensive communication between the components of the system. Blockchain is a promising solution that can bring more robustness and transparency in most vehicular network applications. The adoption of this technology in the vehicular network will provide trust and error reduction through public verification and transparency. Given the importance of traffic data toward improving ITS, a promising technology like blockchain should be exploited.

The main difference between the blockchain-based scheme and the above architectures (i.e., VANET-based Cloud Computing (V-CC) and Vehicular Edge Computing (VEC)) is the decentralization property of the former. The decentralization allows VN entities to cooperate and make decisions independently without trusting each other. Thus, reliance on third-party entities such as control centers and trusted intermediaries will be eliminated.

In addition, the adoption of blockchain in VN will mitigate security threats such as availability and single-point-of-failure vulnerability. Through replication and synchronization of blockchain state across the network, the system can support and address potential security issues (cf.2.3.5). As a matter of fact, road-traffic related services remain available, even in the presence of attackers. That feature coupled with standard security measures such as modern cryptographic techniques will be a major asset in securing VN.

Finally, the immutability property inherited from the blockchain will prevent traffic information altering. That, therefore, allows for accurate auditing and tracking of the road traffic data.

Both centralized and distributed architectures have failed to ensure secure traffic data management. Recently, attempts have been made relying upon blockchain technology. However, blockchain inclusion in VANET remains at its early stages. We classify existing approaches based on how blockchain is integrated within VN into three categories: vehicle-centric, RSU-centric, and cloud-centric blockchains. We also present a typical architecture of each type and discuss the advantages and

disadvantages of each.

3.4.2 Vehicles as blockchain nodes

In this category, the blockchain is enabled at the ad hoc layer, whereby vehicles maintain the blockchain and orchestrate the consensus. Various vehicles supporting blockchain schemes have been proposed seeking to secure VN applications. Alouache et al. [114] proposed a blockchain-based approach for credit payment in Vehicular Cloud Computing (VCC). The proposed scheme aims to ensure secure and transparent collaboration between vehicles while forming a vehicular cloud. The authors claim that their solution alleviates the problem of selfish cars; they also believe that their proposed Bitcoin-based incentive model will motivate vehicles to exchange data and services with other vehicles through V2V communications.

Moreover, Singh and Kim [115] presented a blockchain-based framework for secure data sharing and trust management for Intelligent Vehicles (IVs). This framework intends to securely store the details related to IV communications. In [115], vehicles solve the PoW puzzle, participate in consensus, and maintain a blockchain of their communications history and trust points.

In addition, Shrestha et al. [116] proposed a blockchain-based model for traffic messages and vehicle trust value storage. All vehicles download, store, and update the blockchain. As a result, each car independently manages the complete history of vehicles trust and road traffic messages. The blockchain integrity is ensured by a PoW consensus mechanism orchestrated by the cars. The authors have mentioned the scalability limitations of their scheme and suggested zoning techniques coupled with edge computing as a prospect to reduce block mining load on the vehicles. An implementation is needed to validate their model.

Wagner and McMillin [117] provided a more detailed protocol for a vehicle-based blockchain scheme, wherein vehicles within the same platoon maintain a collaborative blockchain. Their proposed method requires each car to maintain a blockchain updated by downloading the newly joined platoon state. However, although the authors emphasized the importance of verifying road traffic events by vehicles on the road, the proposed scheme is infeasible due to the high latency for a car to join a platoon and synchronize its blockchain with that platoon.

Similarly, Awais Hassan et al. [118] proposed a blockchain-based framework to secure warning messages exchanged between vehicles without the need for RSUs. The proposed scheme considers important road traffic warning messages such as lane change, forward collision warning, and hard braking warning messages. The

authors claim that messages exchanged between vehicles through a blockchain system could help distinguish malicious vehicles. However, no details on the blockchain protocol were provided; hence, it is challenging to assess the suitability of their proposal for road traffic data management.

The above-cited approaches are attractive because of their high decentralization and their ability to provide direct V2V communication without relying on any infrastructure. This last point, of course, minimizes the RSUs deployment and maintenance costs. Nevertheless, such schemes will not be able to accommodate the required QoS for road traffic data management [119]. Due to vehicle mobility, it is difficult to achieve consensus among mobile nodes due to potential communication problems. In addition, blockchain consensus mechanisms are generally resource-intensive (i.e., high computation or communication requirements), which adds a heavy load on vehicles and therefore disturbs their primary task of sensing and disseminating road traffic data [120].

Mostafa [121] attempted to minimize the consensus load on mining vehicles, leveraging on a mini-blockchain scheme [122]. The aim is to prune the blockchain (i.e., remove outdated blocks) to keep the size of the blockchain manageable. However, the latency and computational load issues are still unsolved.

3.4.3 RSUs as blockchain nodes

In different works, RSUs were relied upon as blockchain nodes. In such a configuration, the consensus load (i.e., storage and computation) will be alleviated from the vehicles. That will leave them with the full potential to efficiently sense and forward relevant traffic warnings.

Yang et al. [123] introduced a blockchain-based system to verify and validate collected traffic data while trusting the RSUs. In [123], vehicles upload traffic data to the RSUs that validate it relying on passing cars' trust. The validated data is then securely protected by a blockchain maintained between the RSUs. Furthermore, in [123] a new consensus protocol, named Proof of Event (PoE), was introduced to validate traffic events. Nevertheless, the proposed scheme does not consider potential attacks on the RSUs, which is a strong assumption (cf.2.2.3).

Moreover, Lasla et al. [124] used a blockchain to propose a fully distributed system to mitigate security vulnerabilities and overheads in centralized authentication systems. The proposed solution relies on a set of RSUs that decide whether to admit or revoke vehicles based on predefined rules. Once consensus is reached, a blockchain of vehicle membership status is updated and maintained between the RSUs. Later, when a vehicle receives a message, the authentication can be

performed by the RSUs. This application is interesting, but its feasibility must be proven with a performance study. Furthermore, details regarding the adopted consensus algorithm must be provided.

In the same context, van der Heijden et al. [125] have presented a blockchain that enables revocation of misbehaving vehicles without requiring centralized trust. Their proposed architecture is based on RSUs maintaining a blockchain of vehicles misbehaviour evidence. The misbehaviour authorities will later access the blockchain data for potential punishment (e.g., certificate revocation).

The above-cited approaches will fail to meet the required storage and computation to maintain a blockchain of road traffic because the RSUs do not have the needed resources. Furthermore, enabling decentralized services at the RSUs through a blockchain requires an efficient Big data processing method and optimization techniques that are computation and memory-intensive. Therefore, the RSUs need to be enhanced to support applications such as traffic data management.

3.4.4 Vehicular Edge Cloud (VEC) enabled blockchain

In other approaches, the blockchain had been coupled with the Vehicular Edge Computing (VEC) architecture [126, 127]. In such an architecture, RSUs are supported by edge nodes providing the needed storage and computation to manage a blockchain of road traffic data.

Li et al. [128] proposed a carpooling scheme relying on blockchain-assisted fog computing. The proposed solution relies on a private blockchain to record carpooling processes while maintaining vehicle and user privacy. The authors claim that their proposed solution is secure if RSUs allow fog nodes to be semi-trusted (i.e., they can only probe user data). However, this is a strong assumption and may conflict with the need for blockchain in the first place.

In addition, Kang et al. [129] proposed a blockchain for secure storage of vehicular data. The authors assume that their system enables safe and efficient data storage, relying on RSUs equipped with edge nodes. Multiple RSUs are grouped to form edge clusters that temporarily store vehicular data before uploading it to the central cloud.

RSUs supplied edge nodes for blockchain support is the most feasible and promising architecture among those mentioned above. Equipping RSUs with edge servers gives them the resources to handle vehicle requests alongside the blockchain support. Even though edge nodes have limited resources, they can improve RSUs ability to provide the most relevant data through caching to improve the trans-

portation system. Therefore, VEC enabled blockchain is the adopted architecture in all contributions of the thesis.

3.4.5 Consensus protocols for RSUs as blockchain nodes

This section discusses consensus protocols for RSUs as blockchain nodes. PoW and PBFT are the most adopted consensus mechanisms [113] due to their well-established security guarantees. However, some few works relied upon other consensus mechanisms, such as PoA and PoS. These latter are less secure than PoW and PBFT and require synchrony (i.e., messages are delivered within a limited time frame). However, the RSUs network is not reliable, and such an assertion cannot always be guaranteed (cf.2.2.3). Brand-new consensus schemes were introduced specifically for VN; for example, Proof-of-Event (PoE) [130] and Proof-of-Driving (PoD) [115]. These consensus protocols are not reliable because of lack of security guarantees. In what follows, we review PoW and PBFT-based blockchains enabled at the RSUs layer.

PoW based protocols

Attempts have been made to build a secure, immutable, and decentralized traffic records history relying on PoW consensus. Zhang et al. [131] proposed a blockchain scheme to establish a secure, distributed, and decentralized database of vehicle messages on the Internet of Vehicles (IoV). The main objective is to support traffic announcement messages in a large region. The proposed architecture divides the area into subregions, whereby each region has an auxiliary blockchain that stores message specific to its geographical zone. In addition, the auxiliary blockchains are managed by a parent blockchain that ensures the data consistency between these latter. In [131], vehicles are supposed to be powerful enough to solve the PoW alongside the RSUs. However, mining vehicles raise performance issues, as in vehicles blockchain nodes schemes (cf.3.4.2). Zhang et al. [131] did not provide any performance measures to support the feasibility of their approach.

Other approaches have preferred to leverage exclusively on RSUs or external computation providers (e.g., Edge computing nodes) aimed to alleviate the PoW load on vehicles. For instance, Leiding et al. [132] have introduced a public blockchain to implement VANET services without mining vehicles. The proposed infrastructure relies on smart contracts to deploy important VANET applications in the Ethereum blockchain. These applications concern traffic regulation, vehicle tax, vehicle insurance, and other applications that enforce network rules and regulations. Finally, each car is linked to an Ethereum account and pays a fee

(e.g., ethers) to access the blockchain services. Thus, cars are only Ethereum clients; they do not participate in the consensus. Instead, mining should be done by RSUs or by external computation providers. This approach is practical for service-oriented applications as insurance and vehicle tax payment, where the latency of several days may be acceptable; nevertheless, it is not suitable for traffic data management due to its time-criticality.

The previous works lack a thorough study on the PoW-based blockchains adaptation for traffic data management; they also do not provide performance evaluation of their protocols. Performance metrics such as the required delay to confirm a road traffic message or the blockchain throughput are crucial to validate blockchain adaptability in VN. Therefore, with respect to existing works, more study is needed toward PoW adaptation for real-world VANET scenarios.

PBFT-based protocols

PBFT is the other most relied upon consensus protocol alongside PoW; it is a well-studied and proven correct consensus algorithm [133]. A PBFT-based blockchain requires all consensus participants to be authenticated, which does not conflict with VN system since RSUs are authenticated. Such a blockchain protocol is called permissioned blockchain, and sometimes consortium blockchains (cf.2.3.3) since each RSU protects its interest (e.g., its reputation) in the blockchain network.

Zhang and Chen [134] relied on PBFT to propose a consortium blockchain for data storage and sharing in VANET. The proposed scheme relies on a pre-selected RSUs managed by the TMA, which is fully trusted. The proposed blockchain is orchestrated between those selected RSUs, which validate and forge new blocks of traffic data. However, such a scheme has centralization issues because the same RSUs validate the blockchain data. Furthermore, road traffic event validation won't be efficient if validating RSUs are located far from the event appearance location. For example, if there is a traffic jam at a given zone, this event should be confirmed by close RSUs through vehicles in that specific zone. In addition, the proposed scheme has considerable latency (10 minutes for block confirmation), partly resulting from using PoW for PBFT leader election.

Recently, Firdaus and Rhee [135] proposed a blockchain scheme to secure data sharing in Vehicular Edge computing. A PBFT between RSUs was relied on to reach an agreement between RSUs. A network of 10 RSUs was simulated, and the evaluated metrics were related to network metrics such as packets delivery ratio to the RSUs, packets receiving rate, and MAC/PHY layer overhead.

Instead of PBFT, Kang et al. [129] proposed a consortium blockchain to secure traffic data leveraging on a different voting-based pattern. Authors claim that

their proposed consortium blockchain ensures secure and efficient data storage and sharing in Vehicular Edge Computing and Networks (VECONs). In [129] a new voting-based consensus pattern that works in four rounds of communications was introduced. The proposed consensus pattern relies heavily on the leader, as illustrated in Figure 3.6; that highly exposes the leader to DoS attacks compared to PBFT. Also, PBFT operates one communication round less (cf.2.3.2).

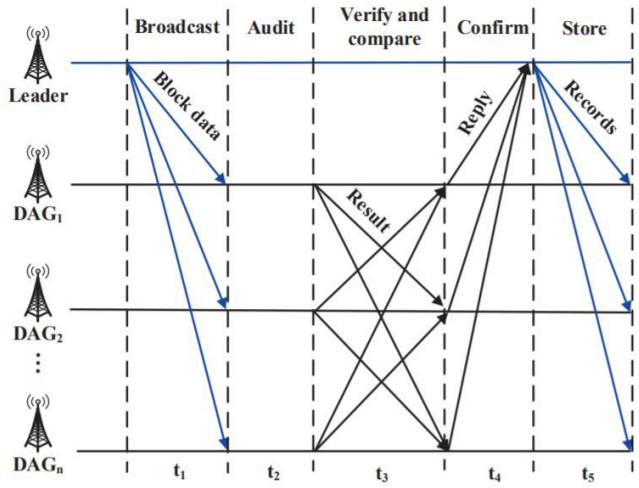

Figure 3.6: The consensus pattern presented in [129], DAG (data aggregator in VECONs by the RSUs)

Most of those consortium-based blockchain schemes do not provide performance details about the consensus protocol, nor how it can be adapted to deal with traffic messages in VANET. From a theoretical standpoint, PBFT performance and security principles have been extensively discussed in the literature [38, 64], but mostly for currency-based use cases such as Bitcoin; however, traffic events differ from Bitcoin transactions. The latter rely entirely on the history (*i.e.*, committed data) for transaction validation, while traffic records need to be approved by the witnesses of the concerned events. It is therefore essential to study the adaptability of widely adopted consensus protocols, such as PoW and PBFT, to build secure traffic records sharing system in VN. More performance studies are needed regarding the throughput and latency, as well as, the size of the blockchain when real-world VANET settings are considered.

Furthermore, we believe that the decentralization and efficiency of existing PBFT-based blockchains can be improved. Events validation efficiency can be improved by dynamically selecting the validators based on event location.

3.4.6 Limitations

Blockchain integration into Vehicular Network (VN) will improve VN applications security. However, blockchain properties such as decentralization, data integrity, and transparency are not directly inherited. Blockchain technology can compromise confidentiality due to data replication between the network participants and its high transparency. In addition, this technology is known to have scalability limitations in handling large amounts of data, which hinders its adaptation into VN.

- **Adaptation** : The adaptability of blockchain in VN is still immature. Most of the existing works only present the basic concept of blockchain-enabled VN; further investigation is still needed:
  - Road traffic data are typically cyber-physical, so their validation must be considered during the blockchain consensus for effective road traffic data validation.
  - How to cope with the massive amount of information such road-traffic data.
  - VANET communications and configurations must be considered.
  - Also, vulnerabilities of vehicles as well as RSUs must be considered and countered.
- **Security**: the blockchain security properties stem from the consensus protocol it uses. For example, the PoW consensus mechanism is based on the computational power of the blockchain network; it guarantees the security of the blockchain as long as the adversary (i.e., faulty/malicious nodes) does not hold the majority of the overall computational power. In such a protocol, the higher the number of RSUs, the more secure the blockchain. For instance, using PoW on a small number of RSUs may not give the same security level as in Bitcoin, which is secured thanks to its high number of miners (cf.2.3.2). Furthermore, although introducing a new consensus mechanism is interesting as in [129], it should be followed by thorough security analysis and guarantees. Some existing works [123, 129] assume that RSUs are trustworthy. However, due to their wide distribution RSUs are vulnerable (cf.2.2.3).

  Therefore, aimed to propose a blockchain-based system for traffic data management, we make no such assumption. Instead, we consider that vehicles and RSUs are potentially malicious at specific limits and evaluate their impact on the robustness of the blockchain.

Note that vehicle privacy-preserving is a challenge for blockchain inclusion within VN. Due to the blockchain's high transparency and data replication, the anonymity property is not guaranteed. Nevertheless, attempts have been made to solve the problem of vehicle privacy in blockchain-enabled VN. Using anonymous identities is recurrent as a solution that would protect vehicle privacy. Research is underway to improve vehicle privacy when relying on a blockchain in VN [113, 136, 137].

- **Scalability**: Blockchain technology, due to its decentralization property, operates without any trust. Each node in the chain must independently verify the system state, implying performance limitations in the blockchain protocol. Numerous solutions have been proposed to overcome this scalability problem (cf.2.3.6). Especially in vehicular networks, some of those approaches have been adapted to blockchain-enabled VN. For example, sharding or clustering techniques have been relied upon to scale blockchain-based Internet of Vehicles (IoV) [131]. Moreover, various consensus protocols have been used to minimize the overall resource costs of the blockchain system and optimize performance in VN [131]. However, these works are still in the early stage and are often centralized or based on strong assumptions or do not cover the actual potential attacks in vehicular networks.

  In this work, we leverage the specificities of road traffic data to minimize blockchain resource costs and optimize performance without sacrificing some crucial properties of the blockchain, such as decentralization and the consensus mechanism's reliability.

- **Performance**: road traffic applications depend on a massive amount of data provided by vehicles. The existing blockchain-based VN cannot cope with such a large amount of data while meeting the delay requirements. Moreover, due to the high level of mistrust in VN, the blockchain performance will get lower. For example, vehicles are subject to attacks, so the data they provide must be verified based on their credibility, which is a complex protocol added to the blockchain protocol. In addition, RSUs cannot be trusted due to their wide distribution and lack of protection. Because of these vulnerabilities, rigorous verification is needed for traffic data. This verification will work as part of the blockchain protocol. Therefore, blockchain-based VN evaluation needs special attention. Metrics such as traffic data validation delay, communication, and storage costs must be assessed and discussed. However, to the best of our knowledge, there is no performance evaluation of blockchain-based VN [112, 120]. Therefore, a benchmarking platform of these solutions is necessary to evaluate and study their feasibility [112][p.14].

  We developed a blockchain simulator to assess the diverse configurations of

our models. Extensive performance studies were conducted and discussed. Attacking vehicles and RSUs and their impact on the robustness of the proposed solutions were also investigated.

3.5 Conclusion

This chapter presented existing vehicular network architectures, classified into three categories: centralized, distributed, and decentralized. Each architecture was defined, and its advantages and limitations in securing traffic data were discussed. The conducted review showed that while the distributed architecture mitigates the security and privacy issues of centralized schemes, it poses other security issues because widely distributed edge servers are subject to attacks and cannot be fully trusted. These distributed servers must be synchronized without trust. Therefore, decentralized architecture steps in to solve this challenge by leveraging blockchain technology.

Throughout the review, we developed decentralized approaches wherein blockchain technology is enabled for Vehicular Network (VN). We structured existing schemes into three categories based on how blockchain is integrated into VN while discussing each approach. The main gaps in the current state-of-the-art of blockchain-enabled VN are security (e.g., RSUs must not be trusted), scalability, and lack of performance evaluation.

This thesis aims to securely and efficiently adapt blockchain for road traffic data management. Two protocols have been proposed: PoW-based and k-replication. An exhaustive performance study was conducted to validate these protocols in real VANET scenarios.

4

Bitcoin-like blockchain for secure traffic records management : Adaptation and Performance Evaluation

Contents

4.1 Introduction

As the first blockchain application, Bitcoin has proven its robustness. Deployed in an open network, Bitcoin has shown high resilience to attacks, making it the most secure open and decentralized database the world has ever known. Bitcoin's security, as the case for any blockchain protocol, is dictated by its consensus mechanism: Proof of Work (PoW).

The reasonable question to ask is: can PoW-based blockchain be adapted to secure traffic data while inheriting all its security properties? What are the challenges, constraints, and performance outcomes?

Bitcoin is designed to allow its clients to maintain a wallet with their Bitcoin balance and in which they can transfer Bit-coins to other clients without third-party intervention. In Bitcoin, each miner apparently and independently verifies all transactions. This verification concerns account balance verification (i.e., if there are sufficient Bit-coins in the account to allow the transfer) and the transaction format (i.e., if a Bitcoin client correctly signed the transaction) [18].

However, traffic records validation is more complex; it does not depend entirely on the history (i.e., the blockchain data). Therefore, the only effective way to assess event relevance is through vehicles that pass near the event's appearance and witness it with their dedicated devices.

In this chapter, we investigate the adaptability of a PoW-based protocol to secure traffic events. The proposed scheme is validated with rigorous study of performance (i.e., throughput and latency).

The remainder of this chapter is organized as follows. Section 4.2 presents VANET architecture enabled a PoW-based blockchain. Section 4.3 details the proposed protocol. Section 4.4 describes the simulation environment. Section 4.5 evaluates the performance of the proposed system. Finally, section 4.6 concludes this chapter.

4.2 Pow-based Blockchain Architecture

This section shows how the blockchain is adapted into VANET for secure storage of road traffic events. It starts with the definition of VANET components' role in the proposed protocol. Then, we present the event validation methodology and discuss the security model.

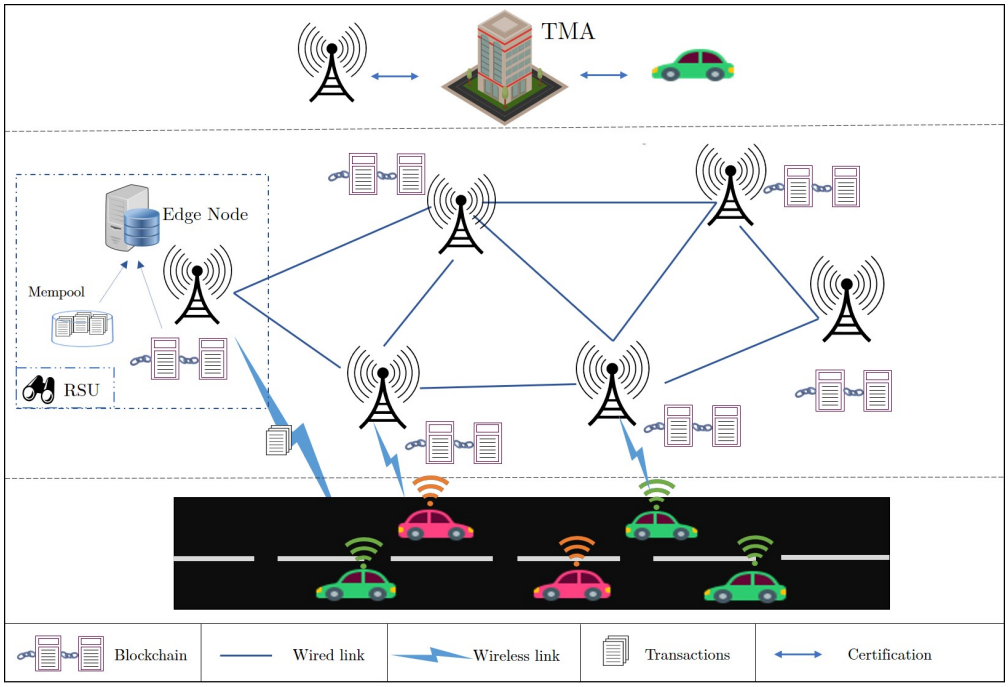

Figure 4.1: Pow-based blockchain architecture

4.2.1 VANET components roles

As illustrated in Figure 4.1, the main components of VANET architecture can be separated into three layers: the ad hoc layer, the RSUs layer, and the last layer is the Traffic Management Authority (TMA) (cf.2.2). In what follows, we discuss the role of each layer adapted blockchain for road traffic events management.

The ad hoc layer defines vehicles supplied with Dedicated Short Range Communications (DSRC) modules, easing their communication with their environment. Vehicles can communicate with other vehicles (V2V) equally with nearby RSUs (V2I) via their DSRC modules (cf.2.2). Besides, vehicles are equipped with smart devices that allow them to collect information about road conditions. For instance, they can record traffic jams, accidents, weather conditions, etc. This data, if proven relevant and trustworthy, will improve the transportation system.

In the proposed protocol, vehicles upload traffic records directly to neighboring RSUs for verification and validation. Later, we detail how uploaded messages are validated. We also assume that vehicles do not solve the PoW puzzle. By doing so, vehicles retain all their capabilities to detect and broadcast traffic data to the RSUs.

The RSU layer consists of RSUs maintaining a Peer-to-Peer (P2P) network between them and a blockchain of traffic events as shown in Figure 4.1. The RSUs are equipped with Edge nodes that provide them with the necessary computation and storage resources to maintain the blockchain. Incorporating the Edge cloud paradigm into the VANET architecture is essential to meet the needed computation to solve the PoW puzzle, as well as the high storage space requirement (cf.3.4.3).

The RSUs play an important role in traffic events assessment. As the reported events may be erroneous, their trustworthiness must be validated to ensure the reliability of the blockchain data. Therefore, we rely on a threshold-based methodology to validate traffic events. More details are given later in section 4.2.2.

The TMA layer represents the Traffic Management Authority (TMA); it is a trusted organization generating cryptographic credentials to newly joining vehicles and RSUs. The provided certificates must be anonymous to protect vehicle privacy. In this work, we simplify the certification process, and assume that generated certificates are anonymous and does not compromise vehicle privacy.

The TMA also monitors the blockchain and inflicts punishment to misbehaviour actions. e.g., the trust authority can identify vehicle misbehaviour through the blockchain and consequently take the appropriate decision such as vehicle certificate revocation to protect the VANET.

4.2.2 traffic events validation

An efficient traffic event validation is crucial for blockchain data reliability. For example, vehicles could report fictitious events intentionally, trying to jam the system, or due to software crash (cf.2.2.3). Therefore, validation of messages coming from vehicles is the first step toward trustworthy and reliable traffic records sharing and storage.

We rely on the RSUs to assess the traffic data. More precisely, after receiving a message reporting a traffic event, a RSU waits for a threshold of confirmation messages from passing cars before considering the event plausible. If necessary, a RSU would request confirmations from nearby vehicles as illustrated in Figure 4.2. A valid road traffic event must be confirmed by a given number of vehicles that witnessed the event. The identities of these vehicles will be included in the valid event and stored in the RSU. Later, validated events will be securely included in the blockchain through the PoW mechanism. Thus, a RSU-level blockchain will be built to realize a decentralized, transparent, immutable, and secure system related to traffic in the VANET. In the following, we will describe the blockchain-based

traffic message storage protocol in detail.

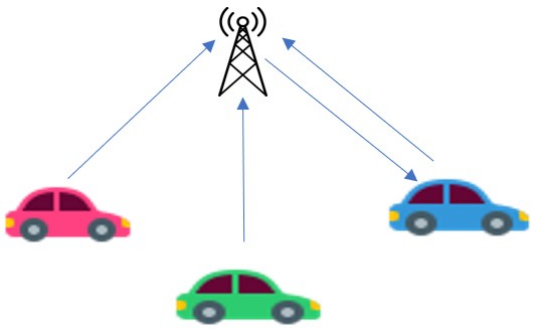

Figure 4.2: RSU-centric events validation

4.3 Protocol Description

This section describes the proposed scheme for secure traffic data validation, storage, and sharing. In Figure 4.3, we present a diagram flow of the proposed model, which consists of 6 steps. We also provide pseudocodes describing block mining and block confirmation processes.

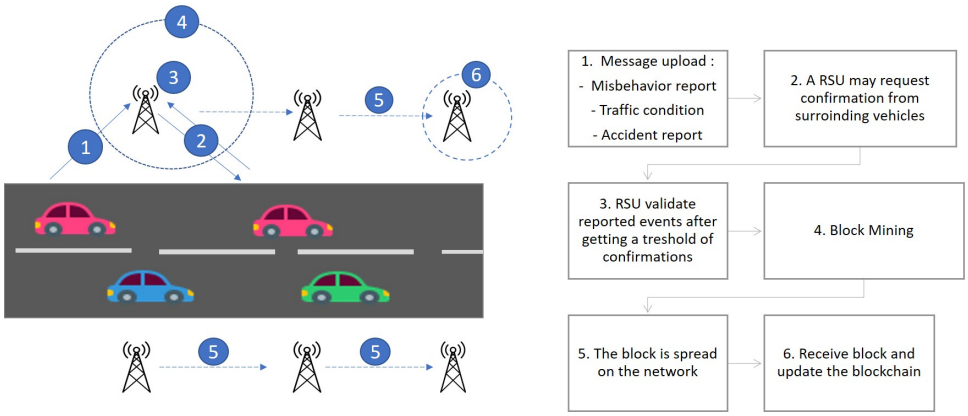

Figure 4.3: Model flow

We consider the scenario illustrated in Figure 4.3, where vehicles upload the road state to nearby RSUs. We recall here that the goal is to establish a secure and reliable database of traffic records. The proposed protocol works in 6 steps, as shown in Figure 4.3.

- The first step is traffic messages collection. This phase consists mainly of vehicles reporting hazardous traffic events sensed through their embedded smart devices. The witnessed events are then sent to nearby RSUs. An event is defined by a type, location, description, timestamp, severity, the signature of the message issuer, and its public key. When received by the RSU, its format is validated (i.e., hash and signature) before being stored into the RSU's *mempool* as depicted in Figure 4.4. The aim is to exploit this event later to optimize the transportation system; however, its accuracy must be assessed first.

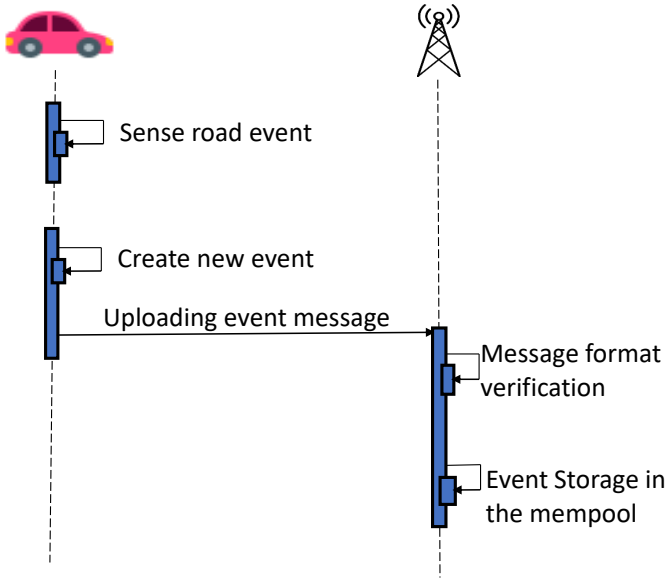

Figure 4.4: traffic messages collection

- So, steps 2 and 3 are devoted to event validation. Once an event message is received by a RSU it will be stored under a pending state, waiting to be validated. The validation is done through a threshold of confirmations received from vehicles. More explicitly, let $V_{i\in\mathbb{N}}$ denotes the vehicles and $M_{i\in\mathbb{N}}$ their associated messages to report a given event e, and thr, the threshold (i.e., the required confirmation messages). For example, if V_0 uploads an event message M_0, the RSUs that received M_0 will wait for thr of $M_{i\in\mathbb{N}^*}$ before considering e to be valid and relevant. RSUs can also request event confirmation from passing vehicles (Figure 4.3, step 2). In case the threshold is not reached after a certain delay, the corresponding event will be discarded from the RSUs' *memepools* as shows Figure 4.5. Finally, a valid event must contain all identities of vehicles that contributed to its validation. Thus, vehicles interaction with the system could be traced and malicious vehicles

(e.g., vehicles that sent erroneous messages) could be identified in posteriori thanks to the Traffic Management Authority (TMA).

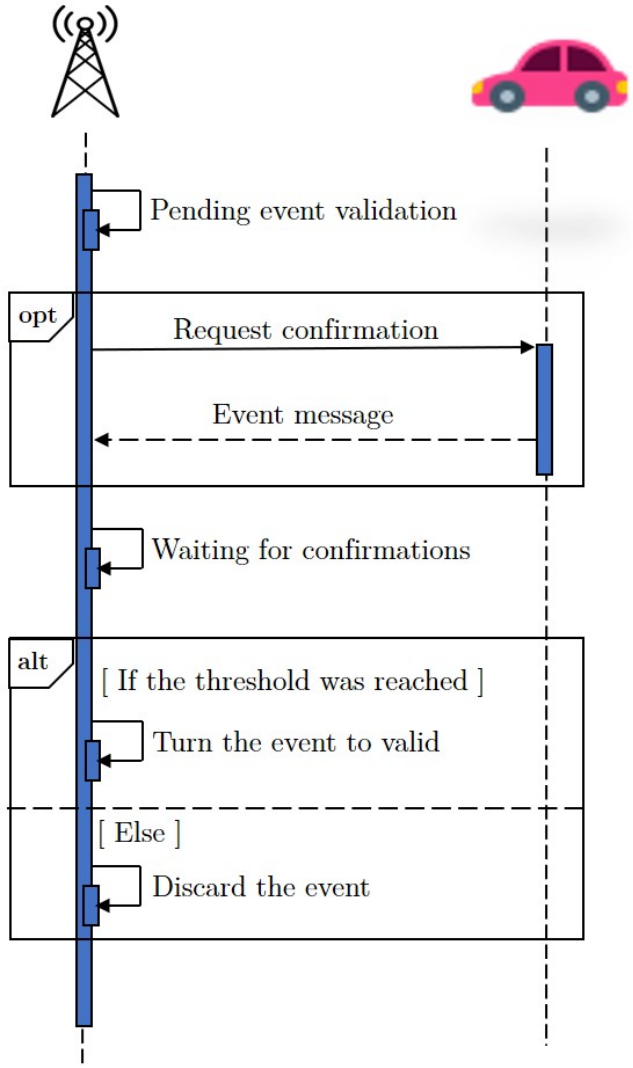

Figure 4.5: Traffic event validation flow

- Step 4 consists of block mining. At this stage, RSUs leverage their computation power to solve the Proof of Work (PoW) puzzle. Algorithm 1 depicts how RSUs mine blocks.
- In step 5, the mined block will be disseminated in the network in a P2P manner. After reception and verification, each RSU adds the block to its local copy of the blockchain and then transmits the block to its peers (except the source). Thus, the block will be distributed to all RSUs.

Algorithm 1 Block Mining

Input :
mempool : $mempool$;
blockchain : $chain$;
peers : $peers$;
block time out : $block_{time}$;
PoW difficulty : d

1: $b \leftarrow createBlock()$
2: $nonce \leftarrow 0$
3: $tries \leftarrow 10000$
4: **repeat**
5: $b.setBlockNonce(nonce)$
6: $b.calculateBlockHash()$;
7: **if** $b.getHash().getLeadingZeros() \geq d$ **then**
8: $break$
9: **else if** $chain.getHeight() \leq b.getHeight()$ **then**
10: $break$
11: **end if**
12: $nonce \leftarrow nonce + 1$
13: **until** $nonce \leq tries$
14: **if** $nonce \leq tries$ **then**
15: $b.shuffleTransactions()$
16: **else**
17: $chain.addBlock(b)$
18: $mempool.update()$
19: **end if**
20: $wait(block_{time})$ ▷ wait for block spreading delay
21: $Repeat()$ ▷ Repeat the mining process

- Finally, step 6 is block verification at reception. When a RSU receives a new block, it verifies if the block has been correctly signed, if the PoW was correctly solved, and finally, if the required threshold of confirmations was reached for each event within the block. Once the block is verified as correct, the RSU stops trying to mine a block with the same height (i.e., order in the blockchain) and appends the block to its local blockchain as shown in Algorithm 2.

Algorithm 2 Receive Block

Input :
mempool : mempool
chain : blockchain
block: block
peers : peers
d : PoW difficulty

```
1: if chain.hasBlock(block.getHash()) then
2:     return
3: end if
4: height ← block.getHeight()
5: if isValidBlock(block) then
6:     stopMinig(height)
7:     chain.addBlock()
8:     chain.update()
9:     sendTo(peers)
10:    mine(height + 1)                    ▷ Start mining the next block
11: end if
12: function isValidBlock(block)
13:    hash ← block.concAttributes()  ▷ Concatenate all block atrributes except block hash
14:    if SHA256(hash) <> hash then
15:        return FALSE
16:    else if hash.getLeadingZeros() < d then
17:        return FALSE
18:    else if not block.reachedThreshold() then    ▷ Test if has enough confirmations from vehicles
19:        return FALSE
20:    end if
21:    return TRUE
22: end function
```

4.3.1 Security model

PoW security is based on the global computing power within the system; it is considered secure as long as adversary miners do not hold 51% of the total computation power [18]—the more the miners, the securer the blockchain. Thus, the more RSUs participate in the consensus, the more secure the blockchain.

For example, in Bitcoin, over 1 million miners make it challenging to gain

51% power over the network. In our context, we might not need to share traffic state further than a specific city; therefore, it seems easier to obtain 51% of the computation power. However, RSUs cooperate within a private network, unlike in Bitcoin, which is deployed in an open network. Therefore, the challenge of 51% attack is still relevant.

Besides the 51% attack, forks increase the advantage of the adversary in the network [38]. Essentially, the higher the number of forks, the less secure the blockchain. Therefore, in the evaluation section, we assess the impact of forks on the overall performance of the adapted PoW-based blockchain.

4.4 Blockchain Simulator

To evaluate the proposed protocol, we implemented the blockchain components: blocks, transactions (which are traffic events), and the PoW as in Algorithm 1. We then set the VANET environment using NS-3 [138, 139], a discrete-event network simulator. A RSU represents a node in NS-3 running an instance of the blockchain. RSUs are positioned following a grid topology such as the average distance between them is $8-11$ km, and each RSU has between 2 to $\sqrt{N}$ TCP connections generated randomly where N is the number of RSUs. Moreover, communications are encrypted using Schnorr signature [140] implemented using the OpenSSL library. Finally, SHA-256 is used as a hash function to hash transactions and blocks.

The presented simulator is used to picture the life cycle of traffic events, from reception, validation to being persistent in the ledger.

4.4.1 Simulation environment

The simulated scenario was set to mimic real scenarios. For example, we have analysed a dataset of alert messages from the city of San Francisco for the year 2019 [141]. The traffic warnings arrival rate is 60 events per day, which is not representative of road traffic data. Indeed, the frequency of traffic alert arrival will increase with autonomous vehicles. Therefore, we vary the events' arrival rate from 1 to 500 event/s following a Poisson distribution to capture low and heavy traffic. In addition, regarding the network, RSUs can be communicating using WiMAX [142], which offers a coverage distance of 15 km with a communication bandwidth of 100 Mbps [142]. Consequently, less than 24 RSUs is enough to cover San Francisco, whose area is $121,4$ km^2. Therefore, the communication speed between the RSUs is set to 100 Mbps and the N the number of RSUs is set to 20.

4.4.2 Evaluated metrics

The evaluated metrics are the average number of events confirmed per second (i.e., the **throughput**(event/s)) of the blockchain, the average delay between traffic event creation time to event confirmation (i.e., the **latency**(s)). Furthermore, the impact of the PoW puzzle difficulty (d), the number of RSUs, and traffic event arrival rate f(event/s) on system performance are assessed. Finally, the blockchain reliability and security were measured through the rate of stale blocks (i.e., frequency of forks) and the longest fork size (cf.2.3.2). Figure 4.6 illustrates the simulation workflow.

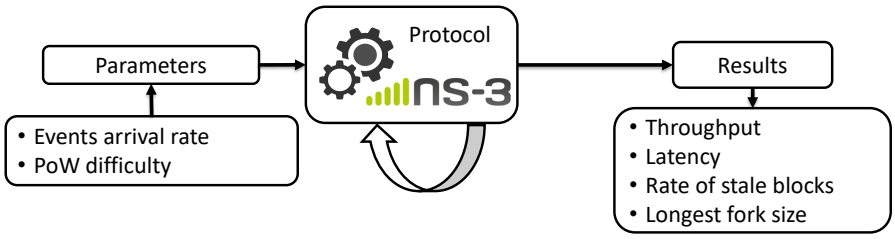

Figure 4.6: Simulation workflow

In this study, the road event size is 800 bytes, as defined in the [134, 143]. The traffic is assumed to be dense enough to validate an event (reach the threshold of confirmations from passing vehicles) within 500 ms. Moreover, $block_{time}$, which corresponds to the waiting time between block mining, is set to 500 ms. That required correct RSUs to wait for enough road events to be included in a block. Also, the block size is limited to 1000 events (0.8 MB). These two last parameters have been set after a multitude of tests. In addition, road traffic events are generated during the first 60s; the simulation stops after the events have been added to the blockchain.

4.5 Evaluation

This section evaluates various instances of the PoW enabled traffic data management protocol using a server with the following 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 plotted results represent end-to-end measurements from all RSUs. The throughput (event/s) is the total confirmed events divided by the simulation time. As for latency, the creation timestamp is subtracted from the confirmation timestamp for each event, then the average delay is calculated.

Finally, each experiment is repeated 10× with different seeds, and the mean is plotted with errors, using the 95% confidence interval.

PoW difficulty (leading zeros)	**Delay** (s)
2	0.0014
4	0.2635
6	91.5003

Table 4.1: PoW solving delay.

We evaluate the PoW puzzle-solving delay by launching multiple instances of PoW with various difficulties (2, 4, and 6). For each difficulty (d), we repeat the simulation 100× to obtain enough granularity. All those 100 PoW puzzle-solving delays are stored in a file through which the RSUs read the PoW delays. Table 4.1 represents the mean of the filtered results from the file mentioned above. Results show that the PoW delay increases exponentially with d.

4.5.1 The impact of events arrival rate and the PoW difficulty on the blockchain performance

Figure 4.7 depicts the system performance with the increasing event arrival rate (f). Results show that reducing the PoW difficulty (d) from 6 to 4 enhances the performance. When d is set to 4, the system can process up to 300 event/s out of 500 event/s generated. And the latency does not exceed $2s$, which is acceptable for advertising announcements such as road congestion, weather condition. Lower performance is measured when $d = 6$ as the throughput does not exceed 106 event/s and the latency can attain $18s$ (9×). This significant difference of performance comes from PoW solving' high delay when d=6. Therefore, d should be reduced for practical latency; nonetheless, doing so may affect the blockchain security.

4.5.2 The blockchain security

Therefore, in Figure 4.8 and Figure 4.9 we assess the impact of the PoW difficulty(d) on the blockchain data reliability. Figure 4.8 depicts the rate of stale blocks in the blockchain. Results show that if $d = 4$, rs can attain 34%. i.e., RSUs waste 34% of their resources (i.e., computation and bandwidth) on processing blocks that won't be part of the blockchain. That is because the lower d, the more the PoW puzzle is solved at the same time by the RSUs. Thus, more than

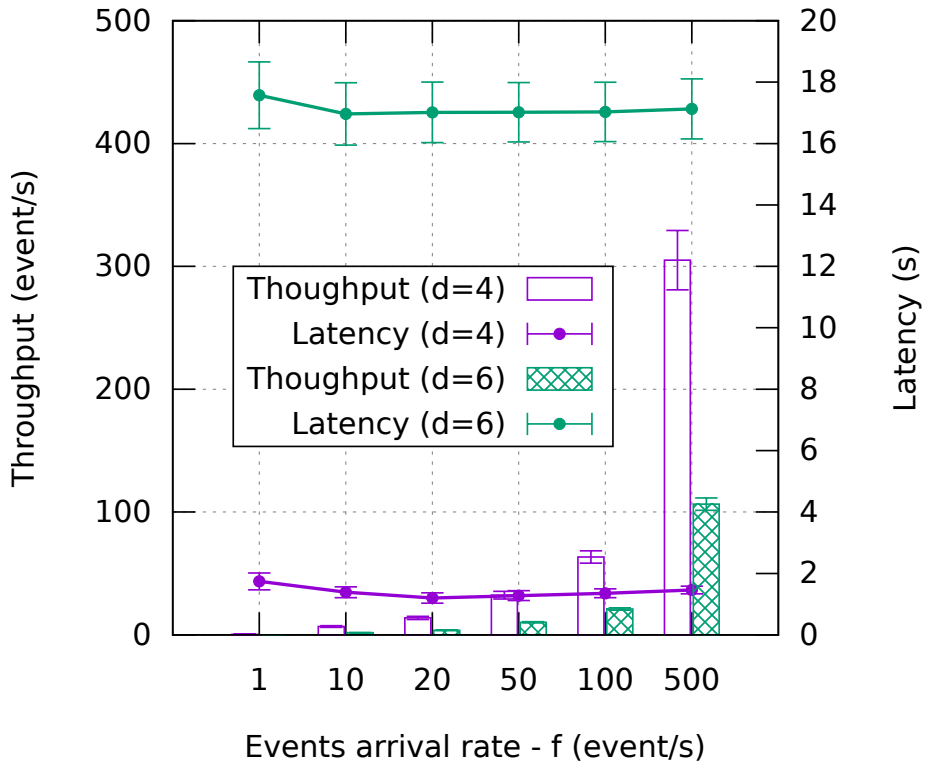

Figure 4.7: Performance (throughput, latency) vs. Events arrival rate-f

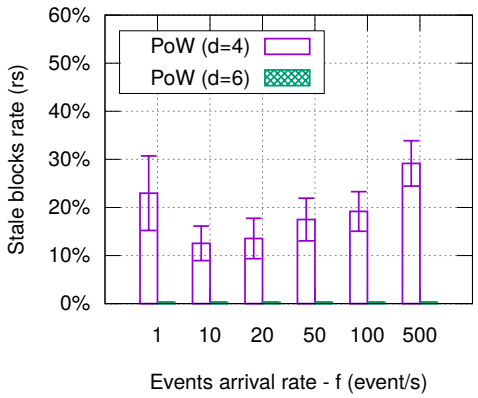

Figure 4.8: Stale blocks (%) vs. Events arrival rate-f

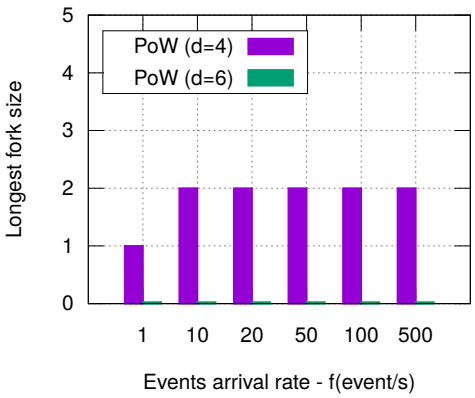

Figure 4.9: Forks size vs. event arrival rate-f

one block is forged for the same height of the blockchain. Hence, leading to an inconsistency in the blockchain, finally, only one block is considered the others are forgotten. The more often this phenomenon happens, the less reliable the

blockchain is. Figure 4.8 also shows that when $d = 6$, blocks are immediately confirmed after being added to the blockchain. Nevertheless, in return, the gained time will be exhausted on PoW solving (cf. Figure 4.7).

To measure the inconsistency in blockchain due to reducing the PoW difficulty (d), Figure 4.9 plots the most extended fork size, i.e., the longest branch that does not belong to the blockchain while varying d. The most extended fork corresponds to the needed oldness (number of previous blocks) before considering a block persistent in the blockchain. The obtained results show that reducing d increases the forks' size because the RSUs is likely to solve the PoW at the same time. For example, in case $d = 4$, the most extended fork is equal to 2 blocks, while for $d = 6$, there are no forks.

4.5.3 The impact of the number of RSUs on the blockchain performance

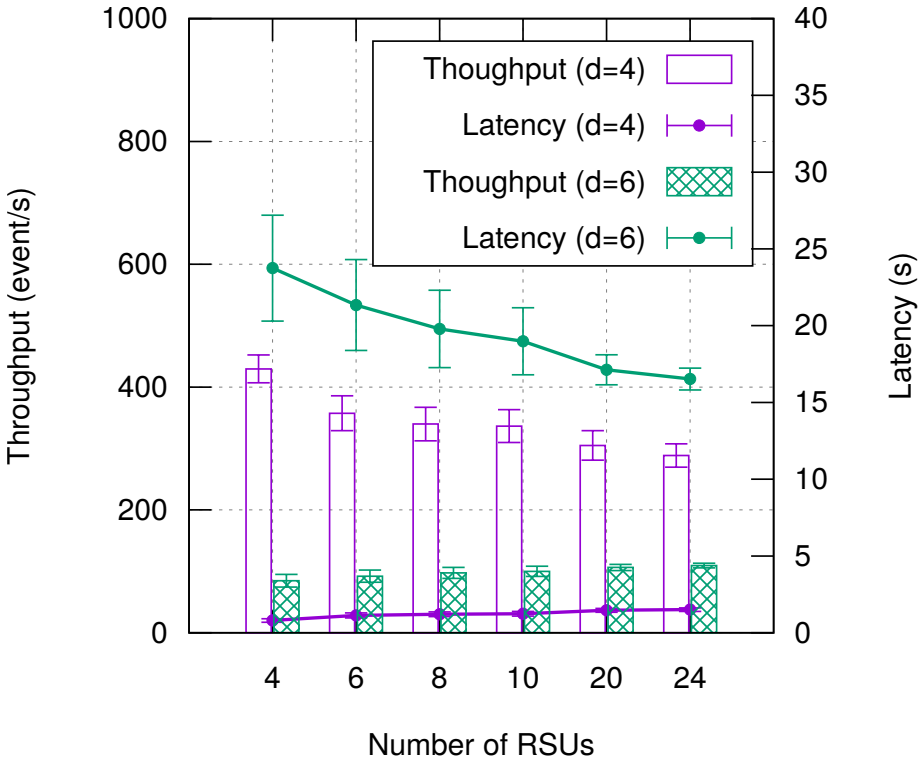

Figure 4.10: Performance (throughput, latency) vs. Number of RSUs (N)

Figure 4.10 shows the impact of the number of RSUs on the system per-

formance. As can be seen, the performance decreases against the increasing number of RSUs when d is low. For example, if $d = 4$, increasing the number of RSUs from 4 to 24 decreases the throughput by 141 event/s, on average. In opposite, if $d = 6$, the blockchain performance gets better when increasing the number of RSUs. For instance, increasing the number of RSUs by 20 reduces the event confirmation latency by $8s$, which is significant; nevertheless, the latency remains high ($> 16s$) when compared to the case $d = 4$ ($< 2s$). Accordingly, the PoW solving delay should be reduced to reach a practical event confirmation delay.

The drop in performance while increasing the number of RSUs, for $d = 4$, is partially due to the block spreading delay, which increases with the network size; nevertheless, it is not the leading cause. Because when the number of RSUs is high, there are more collisions, i.e., RSUs solving the PoW puzzle at the same time. Figure 4.11 confirms that hypothesis, the number of stale blocks increases with the number of RSUs. For instance, increasing the number of RSUs from 4 to 24 induce around 38% of mined blocks to be stale; consequently, higher delay is required to confirm traffic events.

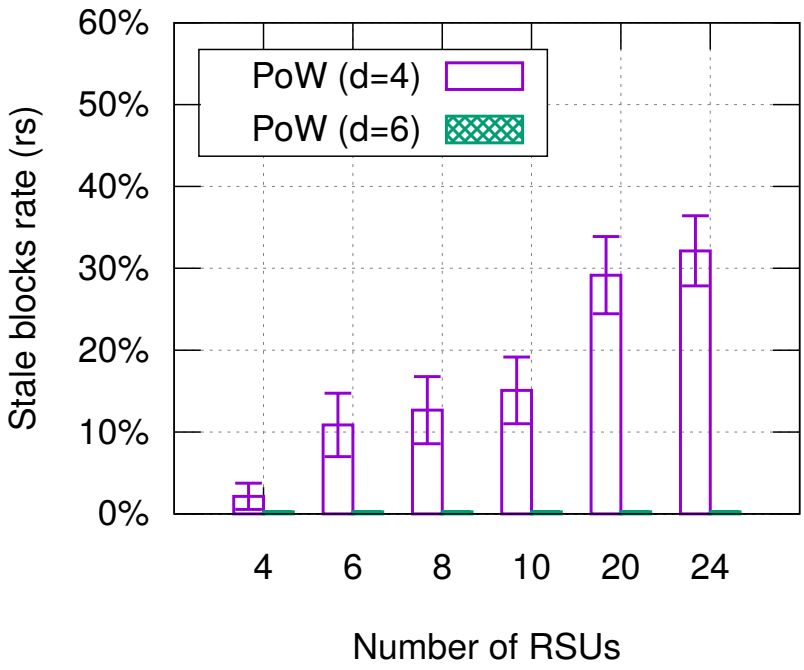

Figure 4.11: Stale blocks (%) vs. Number of RSUs

4.5.4 Results discussion

Results showed that reducing the PoW difficulty would minimize the event confirmation delay; nevertheless, that implies a significant number of forks, which disturbs the blockchain's consistency. Moreover, forks cause a massive waste of computation and bandwidth. On the other hand, a greater PoW difficulty induces a considerable delay due to the PoW puzzle solving. Finally, a trade-off between security and performance must be considered when deploying the proposed protocol.

The proposed scheme provides reliable data of the road traffic state at the RSUs. This data can be exploited by TMA as well as the RSUs to adjust the traffic and provide services to vehicles. This latter should tolerate seconds of latency. Also, vehicles need to implement a lightweight verification system of the information provided by the RSUs.

4.6 Summary

In this chapter, we adapted Proof of Work (PoW) based blockchain for decentralized and secure storage of road traffic events. We considered a real scenario and evaluated different instances of the studied protocol. This study shows how PoW based blockchain can be adapted to enable secure and reliable road traffic history and discusses the performance and security outcomes. The assessed performance metrics are the number of events processed by the blockchain (throughput) and road traffic event confirmation in the blockchain latency. In addition, forks are measured to evaluate blockchain security and reliability. Results showed that PoW could be adapted to build secure and decentralized traffic messages management in VANET. Furthermore, from a performance point of view, results showed that the proposed protocol achieves good performance for traffic efficiency applications by reducing the PoW puzzle difficulty. However, it also was revealed that minimizing the PoW difficulty engenders a significant number of stale blocks, which weakens the security and induces extra resource costs (communication and computation load).

The conducted study in this chapter focused more on the RSUs and the blockchain protocol. In the next chapter, we complete the study by considering the interaction of vehicles with the blockchain. We study the impact of vehicles sending erroneous road events on the blockchain data reliability.

5

Malicious vehicles impact on the blockchain-enabled traffic events management: Performance, Security

Contents

5.1 Introduction

Vehicles are subject to attacks and can spread false information in the vehicular network. If not countered by an efficient traffic data validation system, the erroneous information will be included in the blockchain, which affects the reliability

of the blockchain. In the previous chapter, we introduced a threshold-based event validation method. In this chapter, we study the effectiveness of this latter in countering erroneous messages from vehicles. That is crucial because the data stored in the blockchain must be reliable, i.e., consistent and trustworthy, so that it can be exploited to improve the transportation system.

In addition, we study the impact of the event validation protocol on the overall system performance. Finally, we mimic a real-world scenario that captures vehicle mobility and the V2V communication.

The remainder of this chapter is organized as follows. Section 5.2 presents the threat model, which highlights the considered attacks. Section 5.3 details how events are validated. Section 5.4 describes the simulated scenario. Section 5.5 presents the evaluation results. Finally, section 5.6 concludes this chapter.

5.2 Threat Model

As illustrated in Figure 5.1, we consider the same architecture as in chapter 4. However, the focus is to study the impact of faulty/malicious vehicles on the overall system performance and security.

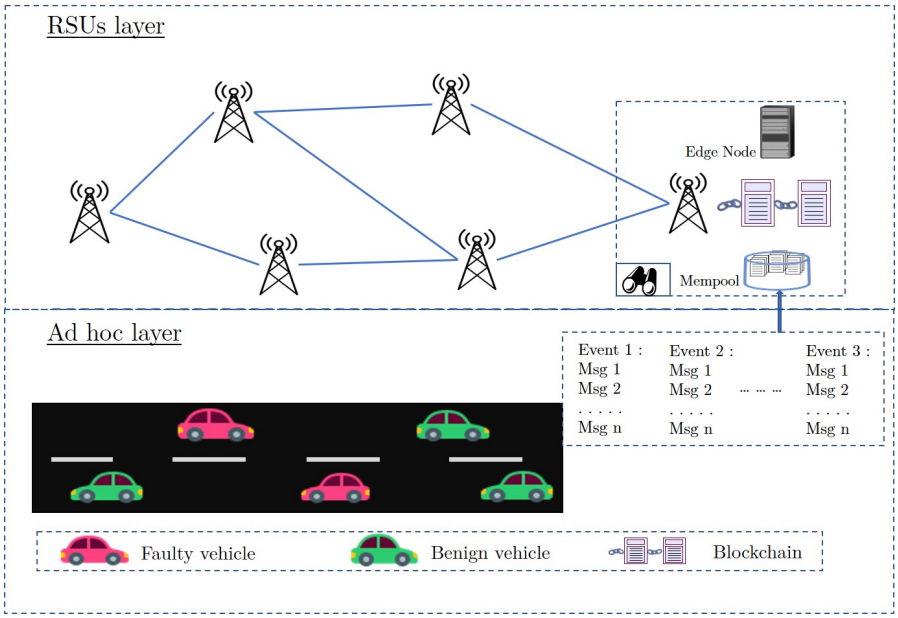

Figure 5.1: Traffic events collection

In general, VANET can be subject to attacks, which can disrupt and degrade the transportation system. Attacks can be perpetrated by authenticated vehicles

or RSUs or by an external entity attempting to infiltrate the network. Therefore, to ensure the security and reliability of the network, both internal and external vulnerabilities must be mitigated. More explicitly, a percentage of vehicles could be compromised (malicious) and deliberately spread false messages in the network. Such attacks are known as Bogus information attack in the literature [22], and they directly affect the integrity of the system. If not effectively countered, false messages related to traffic events can cause significant damage on the road [22]. In the studied protocol, messages from vehicles are validated by witness vehicles and must be immutably included in the blockchain; thus, the effect of malicious vehicles could be attenuated. In the experimental part, we evaluate the impact of faulty vehicles (i.e., vehicles carrying out bogus information attacks) on the system reliability and performance implications.

5.3 Traffic Events Validation

An efficient traffic event validation protocol combined with a secure blockchain protocol provides a robust, reliable, and transparent database of traffic events. The former is crucial to filter erroneous messages from inclusion in the blockchain. This section describes how events are validated by RSUs and prepared for inclusion in the blockchain.

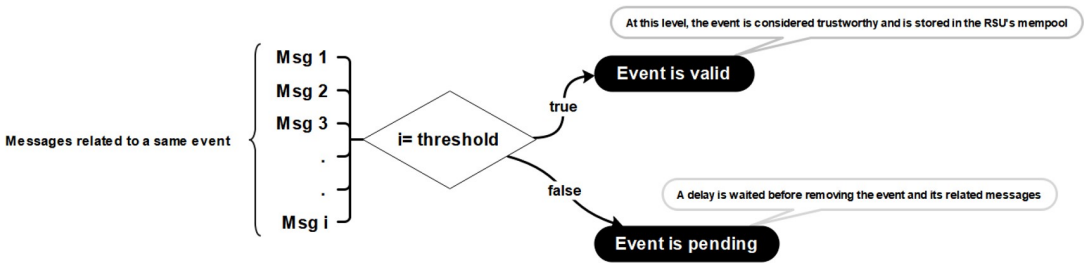

Figure 5.2: Event validation process

As shown in figure 5.2, messages reporting the same traffic event are grouped to form a single event. A given event is alerted by vehicles that witnessed it with their dedicated sensors. For example, if a vehicle is in a traffic jam or on an icy road, it sends an alert message to nearby RSUs. Depending on the density of the road, several vehicles may witness the same event, and thus announce the event to the nearby RSUs. These alert messages confirm the event, so the more of them there are, the more the plausibility of the reported event. Furthermore, all messages confirming the same event are merged, and the public keys of all

vehicles that have uploaded an event must be included in the event. In this way, the traceability of vehicle behaviour toward the system could be achieved, and a vehicle trustworthiness could be measured in posteriori.

Once an event has obtained a defined threshold of confirmations from vehicles, it becomes valid and is therefore included in the blockchain as soon as possible. However, events that have not reached the confirmation threshold remain invalid and have to wait in the RSUs *mempools* in a state of waiting for confirmations. The configuration of the threshold will directly impact the reliability of the blockchain and the event confirmation latency, i.e., the time needed to definitively include an event in the blockchain. For this reason, in the experimental section, we evaluate the impact of the threshold on the reliability of the blockchain in the case of malicious vehicles and its effect on event confirmation latency.

5.4 Simulation Scenario

In this section, we present the simulated scenario and assess the proposed scheme for secure traffic-events management.

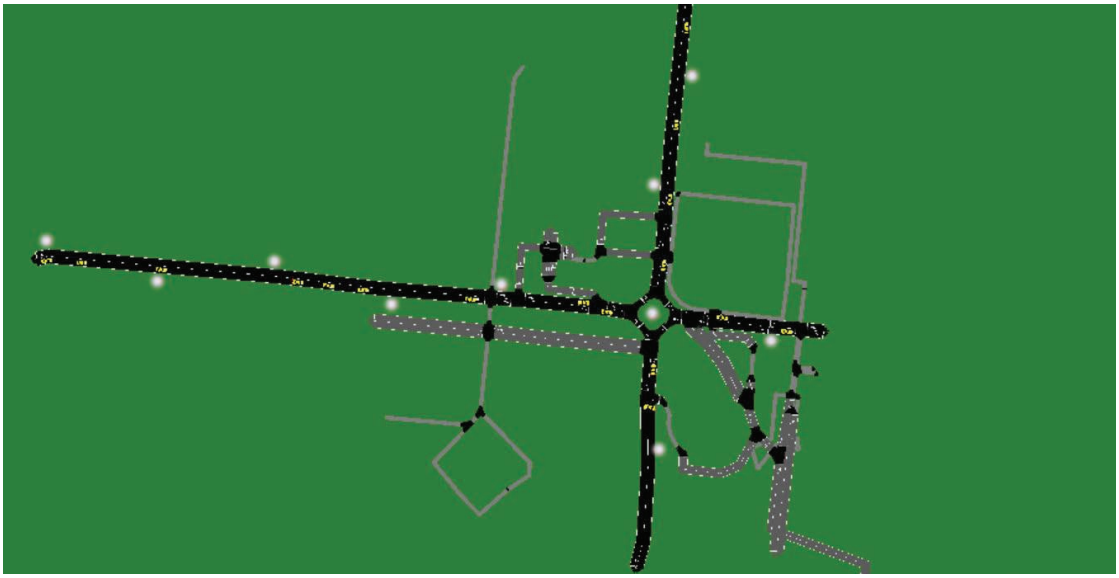

Figure 5.3: SUMO : map of the considered region (the white points indicate the positions of the RSUs)

For our simulation campaign, we have implemented VANET environment in NS-3, a discrete event network simulator. We have created a scenario that tries to recreate a region of the Les Ulis, a municipality in the Île-de-France region. We

have first recreated a set of Les Ulis road junctions, and the associated vehicle traffic, using the dedicated software SUMO (Simulation of Urban MObility) [144]. SUMO is known to create traces that mimic real-world traffic. We have then extracted the mobility traces and integrated them in NS-3, alongside the RSUs. In the considered scenario, there are 10 RSUs and 20 vehicles. A map of the junction can be seen in Figure 5.3.

Vehicles and RSUs networking is rendered through Wireless Access in Vehicular Environments (WAVE) that, in turn, is implemented in NS-3 according to 802.11p @ 10MHz protocol to express V2I communications. The vehicle-to-RSU communications channel data rate is set to 27 Mbps. RSUs are connected by a point-to-point channel, simulating a ground P2P network. All communications are encrypted using Schnorr signature [140], implemented using the OpenSSL library.

We have evaluated the rate of the wrong (erroneous) events included in the blockchain, the impact of malicious vehicles on number of confirmed events and latency of the blockchain enabled secure traffic records storage. We also assess the effect of number of incoming events on event confirmation latency. Some parameters of the simulator have been fixed. For instance, alert messages size 800 bytes as defined in [143]. Also, the frequency of generated messages by vehicles is set to 10 messages per second generated by following a Poisson distribution. A summary of the simulation parameters is displayed in Table 5.1.

Description	**Value(s)**
# of RSUs	10
# of vehicles	20
PoW puzzle difficulty	4 leading zeros
block size	unlimited
message size	800 *bytes*
max # events in a block	unlimited
RSUs link speed : p2p link speed	1 Gbps
warning message generation frequency	10 $event/s$
vehicle average speed	$28.8 - 36$ km/h
Threshold	30% (6)
Percentage of faulty vehicles (%)	[0,5,10,15, 20,25,30]
MAC type	IEEE 802.11p
physical mode	OFDM (27 Mbps rate)
channel bandwidth	10 MHz

Table 5.1: PoW-based protocol simulation parameters

Unless otherwise stated, the threshold is set to 30% (i.e., 30% of vehicles should confirm an event before it is considered valid). And no faulty/malicious vehicles are considered by default.

5.5 Results

The results represent end-to-end measurements from all RSUs. The throughput is examined by dividing the total confirmed transactions by the consensus time, and, for the transaction latency, we subtract the creation timestamp from the confirmation one. Each experiment is repeated for 5× with different seeds, and the mean is plotted. Confidence intervals are omitted for visualization purposes.

5.5.1 Event confirmation delay vs. malicious vehicles

In order to study the impact of malicious vehicles spreading wrong events through the network on the robustness of the blockchain protocols, we simulate instances with attacks by varying the threshold to 5%, 15%, and 30% of the number of vehicles (20), and then measure the number of wrong events that could not be detected by the blockchain protocol.

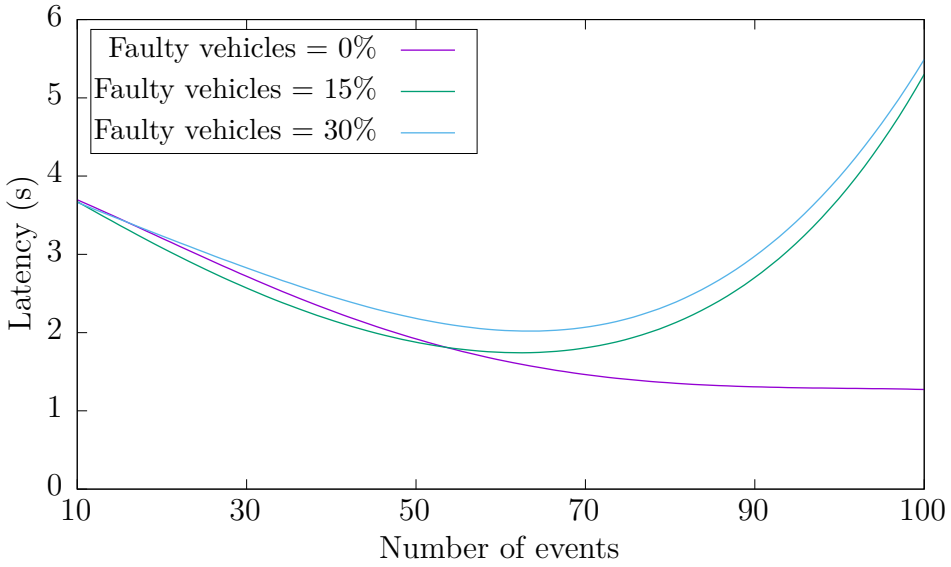

Figure 5.4: Event confirmation delay (latency) vs. Number of generated events

Figure 5.4 illustrates the impact of faulty vehicles on event confirmation latency. The results show that the time to confirm an event increases with the number

of faulty vehicles. For example, in the worst case, where 30% of the vehicles are malicious, the latency for confirming an event is over 5s, whereas it remains around 1.3s when there are no faulty vehicles. This increase in event confirmation latency is because the higher the number of attacking vehicles, the more time is lost in processing erroneous events, which increases the system's workload.

In addition to the extra delay, wrong events will also impact the effectiveness of the blockchain in providing relevant information to improve the transportation system. As a result, countering faulty vehicles is a primary subject that must be dealt with from both a latency, and information correctness points of view.

5.5.2 The effectiveness of the threshold on countering wrong events

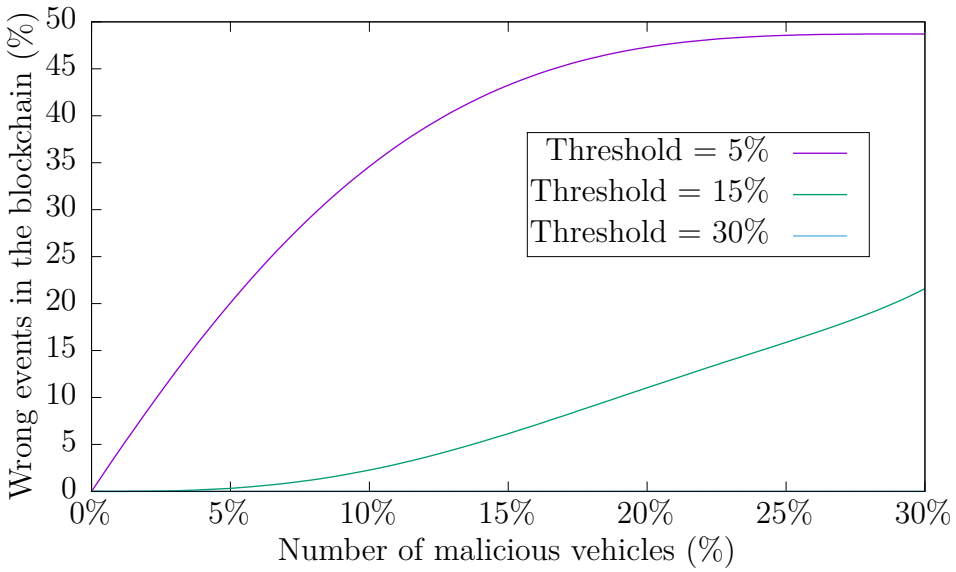

Figure 5.5: Wrong events added in the blockchain vs. Percentage of malicious vehicles in the system

Figure 5.5 shows the effectiveness of the event validation method, i.e., the impact of the set threshold against wrong events. The results show that the higher the threshold, the safer and more efficient the blockchain is in dealing with false events. For example, if the threshold is low, in the worst case (faulty vehicles = 30%), incorrect events could reach 50% of the total confirmed events in the blockchain. However, if the threshold is 30% (i.e., at least 6 of the 20 vehicles are required to validate an event), no erroneous events are confirmed in the blockchain. Therefore, it is crucial to set the threshold high enough to mitigate faulty vehicles'

impact on the blockchain's reliability. However, this could induce an additional latency on the event confirmation time; therefore, it is interesting to assess the effect of a high threshold on the event confirmation time.

5.5.3 The impact of the threshold on the latency

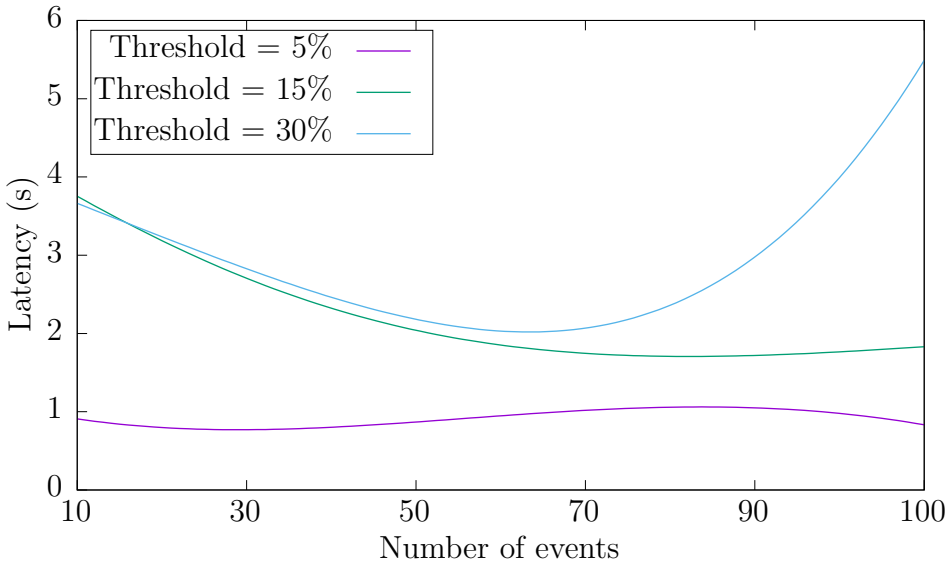

Figure 5.6: Event confirmation delay vs. Number of generated events

Figure 5.6 depicts the impact of the increasing threshold (i.e., the number of vehicles required to validate a given traffic event) on the latency of the blockchain. The plotted results show that a high threshold harms the time taken to confirm the event. For example, if the threshold is 5%, the latency does not exceed $1s$; on the other hand, if the threshold is 30%, the latency is much higher, exceeding $5s$. This increase in the required time to confirm an event is because when the threshold is high, it takes much longer to collect all the necessary confirmations from vehicles to validate an event. At the same time, as previously indicated, the higher the threshold, the more efficient the system is when it comes to handling incorrect events, as shown in the Figure 5.5. Therefore, there is a trade-off between the required security and the desired low latency.

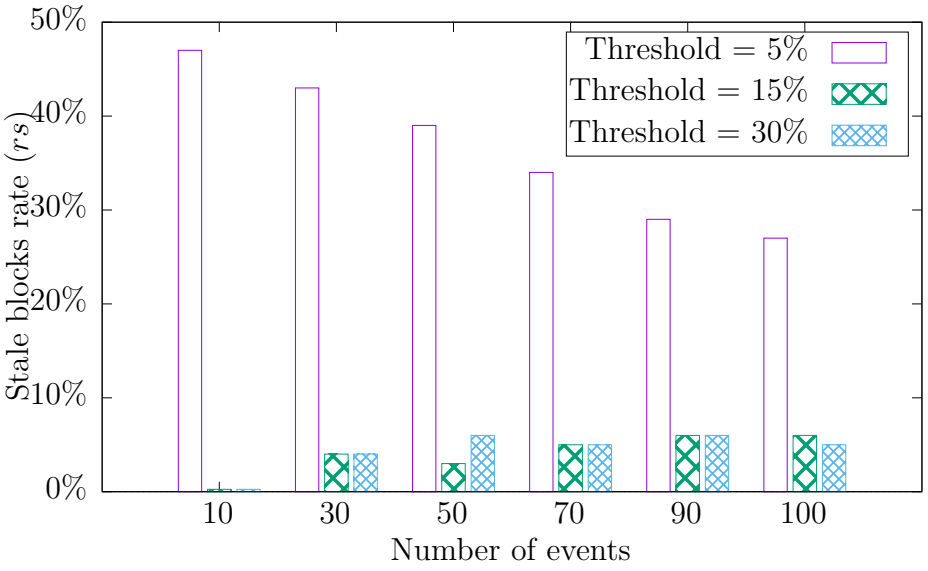

Figure 5.7: The impact of the threshold on the stale blocks

5.5.4 The impact of the threshold on the blockchain security

In Figure 5.7, we measure the percentage of stale blocks in the blockchain while increasing the threshold to validate an event. As the results show, a low threshold generates a massive amount of stale blocks. For example, if the threshold is low 5%, the number of stale blocks reaches 49% of the total blocks in the blockchain. This implies a high inconsistency in the blockchain and affects the overall safety of the blockchain. Therefore, although lowering the threshold reduces event confirmation time, it raises many safety issues, which affect the overall robustness of the blockchain protocol.

5.5.5 Delays on the studied protocol

In order to indicate which stage causes the most delay in the proposed protocol, Figure 5.8 plots the latency for both event validation and the consensus protocol. The event validation delay represents the latency between the event's creation time and the moment when the confirmation threshold is reached. And the consensus delay reflects the time needed to confirm valid events in the blockchain. As can be seen, the highest delay comes from the consensus protocol and represents more than 50% of the total latency for confirming an event. The results also show that the event validation delay increases with the number of events to be validated.

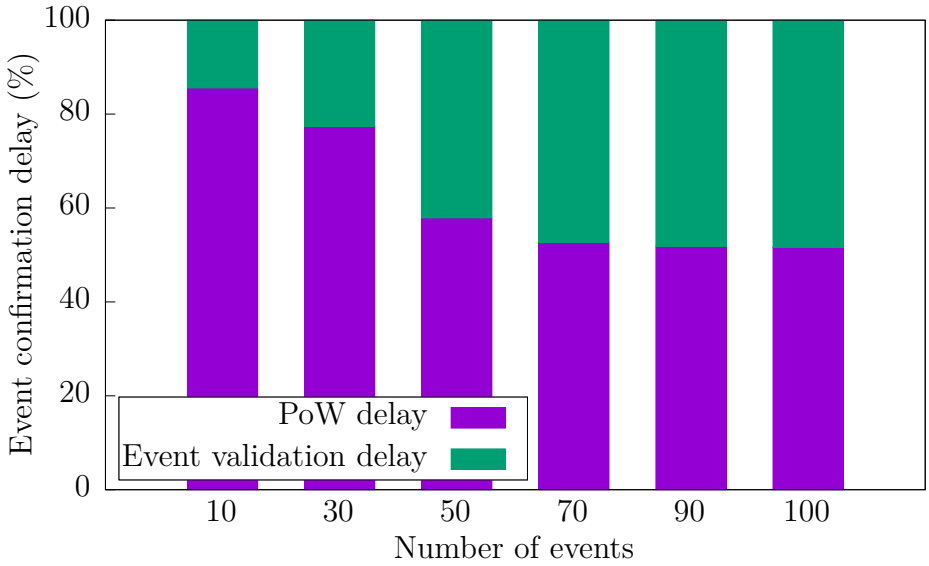

Figure 5.8: Impact of the threshold on the percentage of stale blocks

Therefore, it is crucial to reduce the necessary confirmation threshold without compromising the blockchain security.

5.6 Conclusion

This chapter evaluated the impact of malicious vehicles on blockchain performance and reliability. The simulation results showed that vehicles diffusing incorrect events in the network increase the latency of events confirmation in the blockchain. In addition, a stronger event validation model (e.g., increasing the required threshold of confirmations) would reduce the percentage of erroneous data in the blockchain and improve its reliability. However, that will induce additional delay on the overall latency. Results also showed that relaxing the threshold implies a high percentage of stale blocks (i.e., forks), affecting the security of the blockchain. In addition, results showed that PoW induces the highest delay in the overall system. In the next chapter, we build on a different consensus while taking advantage of the event traffic validation characteristics to minimize the latency.

6

PBFT-based blockchain adapted for secure traffic events management : K-Replication protocol

Contents

6.1 Introduction

The previous two chapters were based on the Proof of Work (PoW) consensus mechanism, a highly decentralized protocol. Nevertheless, a lower delay is needed to reach more time-critical applications. Therefore, in this chapter, we rely on PBFT, a voting-based consensus mechanism. PBFT achieves low latency for a handful of consensus participants. In this work, we leverage the validation properties of traffic events to maintain the system's decentralization while reducing the overall system delay. In addition, we introduce the notion of micro-transactions to minimize the communication delay and storage of the blockchain. The goal is to provide a secure and verifiable history of road traffic data at the RSUs while achieving lower delay and resource costs.

The rest of this chapter is organized as follows. Section 6.2 describes the proposed scheme in detail. Section 6.3 presents the simulation environment, evaluates the protocol's performance, and gives a comprehensive comparison with similar works in the state-of-art. Section 6.4 discusses the results while section 6.5 concludes this chapter.

6.2 Protocol Description

We aim to propose a scalable, secure, and decentralized database for traffic events in VANET. Our goal is to achieve high throughput, low transaction confirmation time (latency) while minimizing the storage and communication overhead. We start with section 6.2.1 that defines the micro-transactions concept. Then, section 6.2.2 presents the blockchain data model. Next, section 6.2.3 describes in detail the proposed scheme. And Finally, section 6.2.4 discusses the security of the proposed protocol.

6.2.1 Micro-transactions

A micro-transaction is a truncated transaction, which contains only the minimum to describe a traffic event. For example, let's suppose that a transaction includes the following details (hash, nature of event, description, location, signature, source, creation time). The associated micro-transaction will only contain (micro-hash, hash, nature of event, location) as depicted in Figure 6.1. The *micro-hash* corresponds to the hash of the micro-transaction; note that it is essential to verify its integrity. To summarize, a micro-transaction is a valid transaction that contains

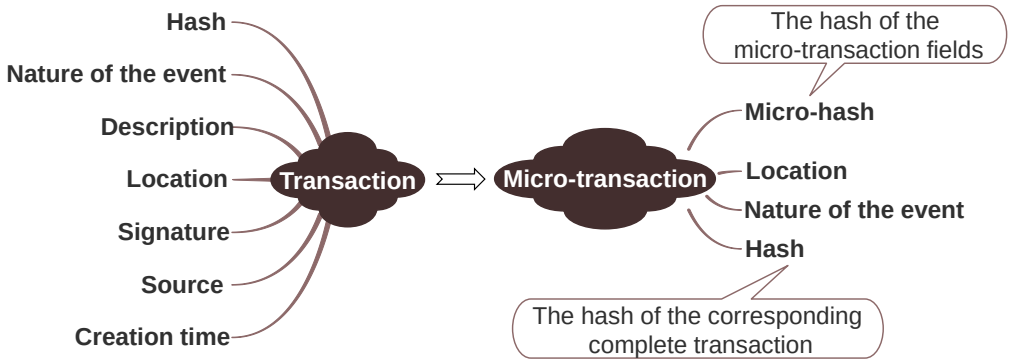

Figure 6.1: Transaction versus micro-transaction

enough data to help a RSU decide whether to request the complete transaction. We refer to a block containing only micro-transactions as a micro-block. The proposed blockchain will be a combination of micro-blocks and complete blocks.

6.2.2 Data model

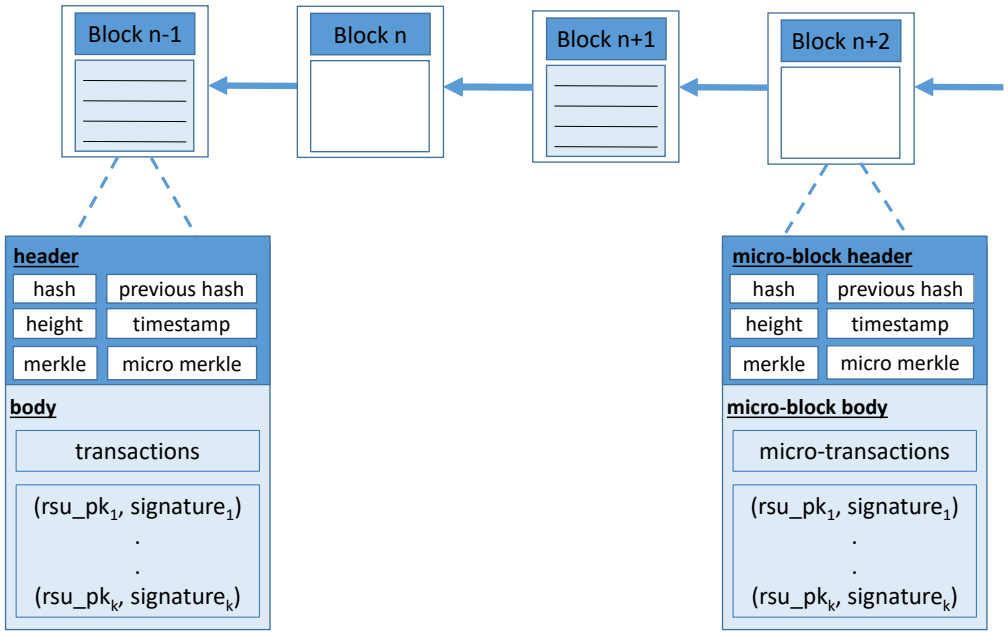

Figure 6.2: Chain of blocks and micro-blocks.

The proposed solution uses the same data structure as the original Bitcoin paper [18]: a list of chained blocks, where each block contains a set of transactions.

However, a new field, *micro merkle*, which represents the Merkle root of micro-transactions, is added to the block. This field ensures the integrity of the micro-transaction. Figure 6.2 shows how complete blocks and micro-blocks coexist.

Historical data is not always needed to approve a road traffic event. For example, verifying an accident's accuracy does not require a history of accidents. Therefore, it is not necessary to permanently store the complete blockchain everywhere. This justifies the usage of micro-transactions to control block replication. Note that the replication must be sufficient to ensure the system's resilience.

With micro-transactions, a RSU may store only the micro-block in which the identities of the corresponding full block holders are defined. Thus, transactions can easily be requested if needed.

6.2.3 Protocol description

This section provides details on the main processes of the proposed protocol: system initialization, block creation, and block verification.

System initialization : initially, before joining the network, the Traffic Management Authority (TMA) has to certify the network participants, i.e., RSUs and vehicles. The certification process consists of generating a pair of private and public keys using the Schnorr signatures [140]. Nevertheless, other asymmetric cryptographic algorithms such as ECDSA could be adopted too.

Block creation : the main difference between blockchain-based protocols is how a new block is created. This process mostly involves the whole network, such as in Bitcoin or a restricted group, trying to reach consensus, as is the case for permissioned blockchain platforms. In the proposed model, the consensus works by rounds. In each round, an elected leader forges the next block. Note that the leader election is an important part of the consensus process. In the literature, we can cite various approaches such as PoW (solving a cryptographic puzzle), Trusted Execution Environment (TEE) [44], and Round-robin as in Tendermint [62]. We consider both the PoW for leader election (PBFT-PoW) and the simple Round Robin (PBFT-ROBIN) and discuss them in the evaluation section.

The algorithm 3 describes the block creation process. It includes the consensus leader election, new block creation, block validation by a consensus protocol, and block confirmation in the blockchain. For more details, the block creation workflow is described in the following six steps:

1. A RSU is elected as consensus leader, relying on the Round-robin mechanism.
2. The leader creates the next block using the traffic events in its *mempool*

respecting some conditions. For example, fixed block size should not be exceeded. Also, transactions within a block must be related to events in the same area (i.e., only events occurring at close geographical positions are collected in the same block). If there are not enough transactions in *mempool*, a RSU has to wait for a defined delay t_{block} before creating the block. This delay should be chosen so that the RSU does not wait too long for a sufficient number of transactions in the *mempool* (which may not come). At the same time, t_{block} should not be too short, so as not to rush towards the consensus without reaching the maximum block size.

3. To define the consensus group (i.e., block validators of the newly created block), we consider the geographical distance between the RSUs and events' appearance locations. We assume that the closer the validators are to event appearance location, the more plausible the validation process will be. With that in mind, for each RSU, rsu_i, we associate a rank, $rank_i$ based on the block. As so, the closer the RSU is to an event, the better its rank. In doing so, we get closer to a realistic context concerning event validation in VANET. Let's denote by $e_{i\in\mathbb{N}}$ the traffic events packed in a newly created block b. The rank of a given RSU, rsu_i is computed as follows:

$$rank_b^i = \frac{\sum_j d(e_j^b, rus_i)}{N^b}$$

where b denotes the newly created block; $d()$ calculates the Cartesian distance between the location of the event and the rsu_i position. And N^b represents the number of events in block b. Once the RSUs are ranked, the next step is to select the highest ranked $k-1$.

4. Next, the leader RSU initiates the consensus by sending a *pre-prepare* message, including the newly created block, to the $k-1$ selected RSUs. Doing so increases the events' trustworthiness. The closer the RSUs are to the event location, the more important their chances to adequately verify the relevance of this latter.

5. The $k-1$ nearby RSUs and the leader validate and agree upon the block during the consensus phase. This latter should contain the identities and signatures of the consensus participants. Hence, easing the public verifiability of the block. Finally, the k RSUs that participated in the consensus append the block to their local copy of the blockchain.

6. If the protocol is set to Full-replication, a full block is spread out through the network to non-consensus RSUs. Differently, in the case of micro-transactions, a micro-block is sent instead of a full block. Consequently, for

each round of the consensus, only k RSUs store the complete blocks, whereas non-consensus $(n-k)$ keep the associated micro-blocks as illustrated in Figure 6.3.

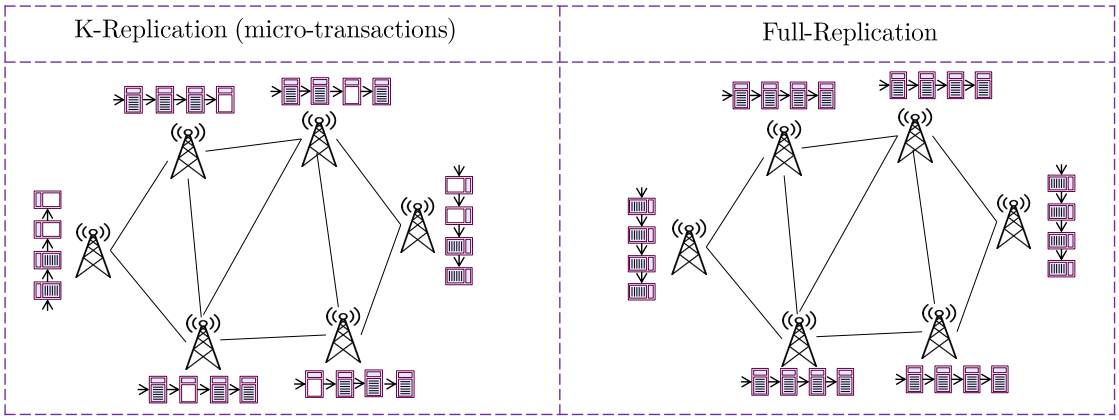

Figure 6.3: k-Replication versus Full-replication ($RSUs = 6$, $k = 4$)

Block verification : besides block creation, block verification is no less important; it ensures the blockchain's reliability and transparency. As illustrated in Algorithm 4, the first step of block verification is to check if k RSUs have correctly validated and signed the block. Block signers are easily accessible since RSUs' public keys are ordered and known in advance; thus, only a bitmap referring to their order is stored. That avoids storing public keys in the block and hence minimizing block size. The next step is to verify that the validators have correctly signed the block's hash. If it is the case, the hash should be re-calculated and compared to the block's actual hash as demonstrated in Algorithm 4.

6.2.4 Security model

By design, blockchain technology raises a trade-off between decentralization, security, and scalability [145]; this is known as the blockchain trilemma. Therefore, it's interesting to balance the security requirements and expected performance (i.e., throughput and latency). The proposed scheme relies on PBFT as a consensus, where a group of k RSUs validates and agrees on the next block. PBFT can support up to $(n-1)/3$ malicious nodes, where n is the number of consensus participants [19]. Therefore, our model can operate under $(1/3)k$ malicious RSUs. The bigger the k, the more secure the system and the poor the performance is. We focus on balancing the security, resource costs, and the system's performance.

Algorithm 3 Next Block Creation

Input:
$prevHash$: previous block hash
t_{now} : timestamp
$height$: blockchain height ▷ number of the last block in the blockchain
$rsu_{1,2...k-1}$: dynamically selected RSUs
t_{block} : block timeout
bs : fixed block size
pks=$pks_{1,2,..n}$: RSUs' public keys
pk : public key

1: **for** each *round* **do**
2: leader = ($round$+ $height$) modulo n ▷ Leader Election
3: **if** $pk = pks_{leader}$ **then**
4: $events = mempool.getEvents(bs)$
5: **if** $events.size() < bs$ **then**
6: $wait(t_{block})$
7: **end if**
8: $events = mempool.getEvents(bs)$
9: $block = Block()$ ▷ Creating new Block
10: $block.setEvents(events)$
11: $block.setMerkleRoot(events)$
12: $block.setMicroMerkle(events)$
13: $block.setCreationTime(t_{now})$
14: $block.setSource(pk)$
15: $block.setBlockHash(SHA256(prevHash||block.getMerkleRoot()$
$||block.getMicroMekle()||t_{now}))$
16: $pbftThreePhases(rsu_{1,2...k-1}, block)$ ▷ PBFT three phases of consensus
17: **if** valid PBFT **then** ▷ If PBFT three phases are correctly achieved
18: **if** protocol is micro-transactions **then**
19: $broadcastMicroBlock(block.getMicroBlock())$
20: **else if** protocol is full-replication **then**
21: $broadcastBlock(block)$ ▷ Broadcast full block
22: **end if**
23: **end if**
24: **end if**
25: **end for**

Micro-transactions do not affect the number of supported faulty/malicious RSUs. k RSUs validate blocks through PBFT; as long as less than $(1/3)k$ are

Algorithm 4 Block Verification

Input *block* : Block
Output isValid : boolean variable

1: $pk_{1,2...,k} = block.getBlockHolders()$ ▷ Get block holders public key
2: **for** pk in $pk_{1,2...,k}$ **do** ▷ Verify that the block was signed by each block holder
3: **if** not $VerifySig(block.getSig(pk), pk)$ **then**
4: isValid = **FALSE**
5: **return** isValid
6: **end if**
7: **end for**
8: $block_{merkleRoot} = computeMerkleRoot(block.getEvents())$
9: $block_{microMerkle} = computeMicroMerkle(block.getEvents())$
10: $block_{header}$= $createHeader(block_{merkleRoot}, block_{microMerkle}, block.getSig(),$
$block.getSource(), block.getCreationTime())$
11: result= $SHA256(block_{header})$; ▷ Calculate the hashed value
12: **if** result = $block.getBlockHash()$ **then**
13: isValid = **TRUE**
14: **else**
15: isValid = **FALSE**
16: **end if**
17: **return** isValid

not compromised, the availability of complete blocks is ensured. Micro-blocks are only spread to the $n-k$ that did not participate in the consensus. Therefore, partial replication has no impact on the security model. In the next section, we assess the impact of the replication factor k on the latency, throughput, bandwidth usage, and storage cost.

6.3 Protocol Evaluation

The proposed scheme is implemented and plugged in the presented simulator in chapter 4 to study its performance. Also, the simulation has been conducted on the same machine as in previous chapters.

The change in the simulation environment is that f, the events arrival rate, is increased to vary from 200 event/s to 5000 event/s to cover hectic traffic. In addition, k is introduced as the consensus group size, and its impact on the performance and resource overhead is measured. Moreover, the block size, bs, is assessed to get the best configuration that maximizes the system performance. Finally, the

network latency is varied to emphasize the effect of micro-transactions on the system performance. Table 6.1 summarizes the simulation parameters.

Description	Value(s)
N : # of RSUs	20
k : consensus group size	$[4, 7, 10, 13, 16, 19, 20]$
f : events arrival rate	$[200, 500, 1000, 2000, 3000, 4000, 5000]$
transaction size	800 bytes
micro-transaction size	300 bytes
bs : max # of transactions per block	$2000, 5000, \infty$
$speed$: p2p link speed	100 Mbps
$latency$: p2p link latency	1 ms, 10 ms, 20 ms
t_{block} : block timeout	500 ms

Table 6.1: Simulation parameters

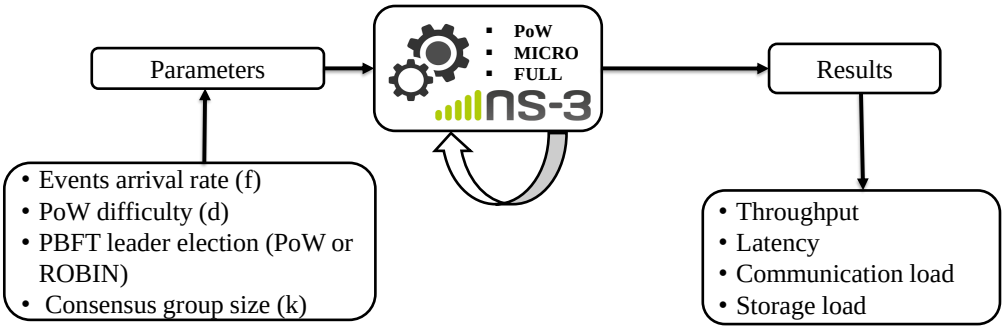

Figure 6.4: Simulation workflow

The simulation workflow is depicted in Figure 6.4. As can be seen, alongside the adapted PoW-based protocol (PoW) proposed in chapter 4, we extend the blockchain simulator with the k-Replication protocol which has two configurations : Full-Replication (FULL) and Micro-transactions (MICRO). The proposed protocol is assessed by computing several performance indicators such as the throughput of the blockchain, its latency, and robustness. In addition, a performance comparison between the k-Replication protocol and the PoW-based blockchain has been made and discussed.

We start this section by the impact of block size (bs) on the performance. By this latter, we refer to the number of transactions validated per second (i.e. throughput), as well as the average time required to confirm a transaction (i.e. latency).

6.3.1 The impact of block size on the system performance

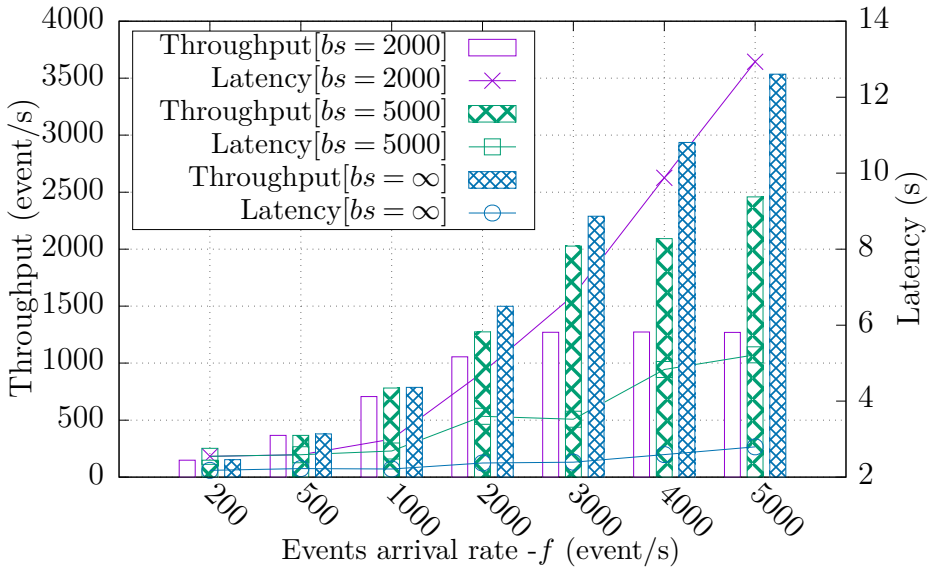

Figure 6.5: The impact of block size (bs) and the events' arrival rate (f) on the performance (i.e., latency and throughput).

To determine the best setting for the block size, in Figure 6.5 we plot the impact of the size of a block (bs) and events arrival rate (f) on system performance (i.e., throughput and latency). The results show that $bs = \infty$ achieves the best performance compared to other configurations where bs is set at 2000 and 5000. For example, when $f = 5000$, the measured event confirmation delays are : $3s$, $5s$ and $12s$ for $bs = \infty$, $bs = 5000$ and $bs = 2000$ respectively.

That is justified by the high speed of the network (100 Mbps). Thus, sending a heavy block does not induce significant latency. Moreover, even if an RSU has to wait for $t_{block} = 500ms$ the required number of events before proceeding to block creation, the results show that increasing bs leads to better performances. Indeed, after t_{block}, the RSU forges the block anyway, even if it is empty.

6.3.2 Effectiveness of micro-transactions

This section evaluates the efficiency of micro-transactions in terms of throughput and latency. Two configurations of our protocol are assessed; in the first, the entire blockchain is replicated across all RSUs (FULL), and in the other, only to RSUs that participated in the consensus. Non-consensus RSUs will only store micro-blocks (MICRO). We vary the number of RSUs participating in the consensus

(k) and set $bs = \infty$ (since this is the optimal parameter when $f = 2000$) and the network latency to $10ms$.

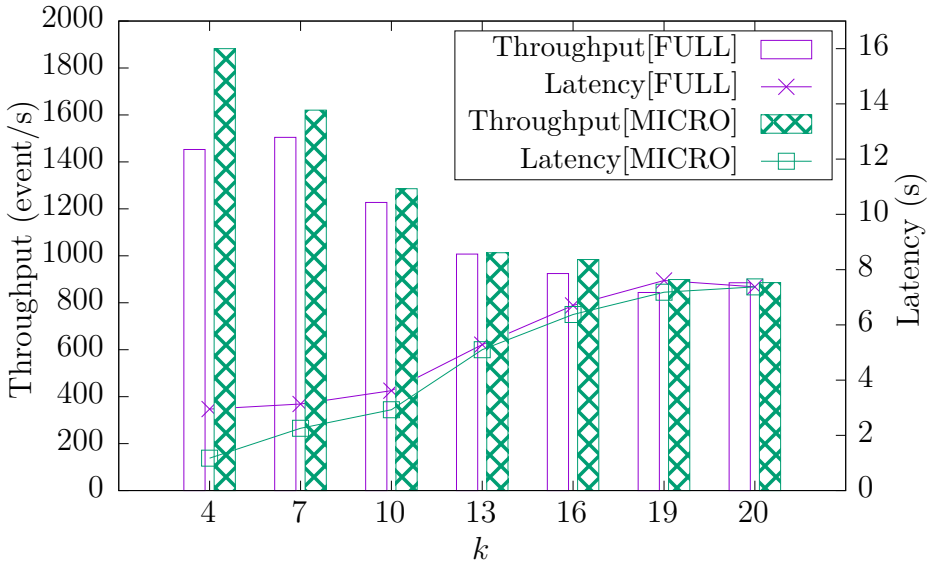

Figure 6.6: The impact of micro-transactions on the replication (k), network latency $= 10ms$

As can be seen from Figure 6.6, results show that when k is equal to 4 among 20 RSUs, the throughput of the “FULL" version is less than the “MICRO". That is because fewer communications are made between the validator RSUs when micro-blocks are exchanged. Moreover, fewer 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 validate 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.

Figure 6.7, and Figure 6.8 depict the gain respectively in terms of latency and throughput when adopting the micro-transactions concept, against the network latency (*latency*). Results show that the effectiveness of using micro-transactions

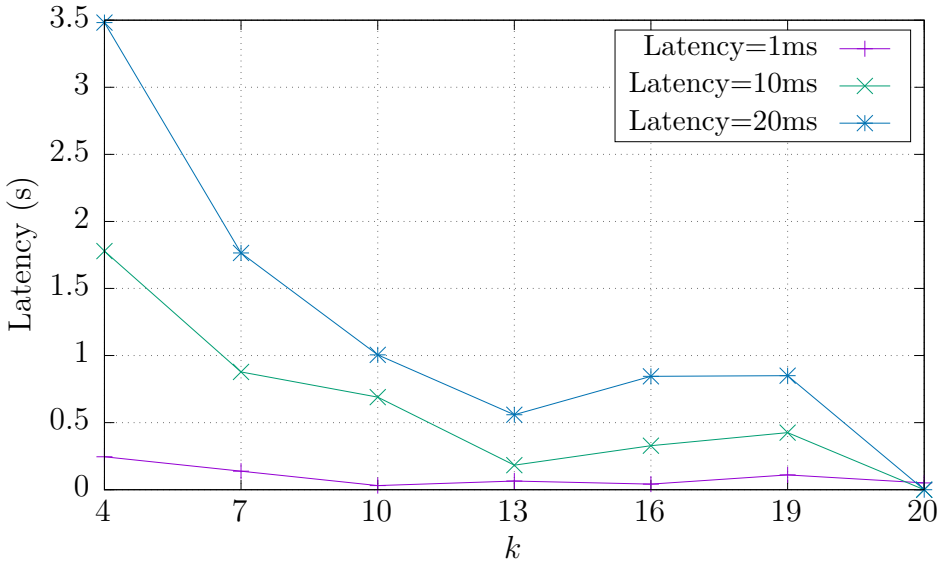

Figure 6.7: The impact of the network latency on the effectiveness of micro-transactions in latency

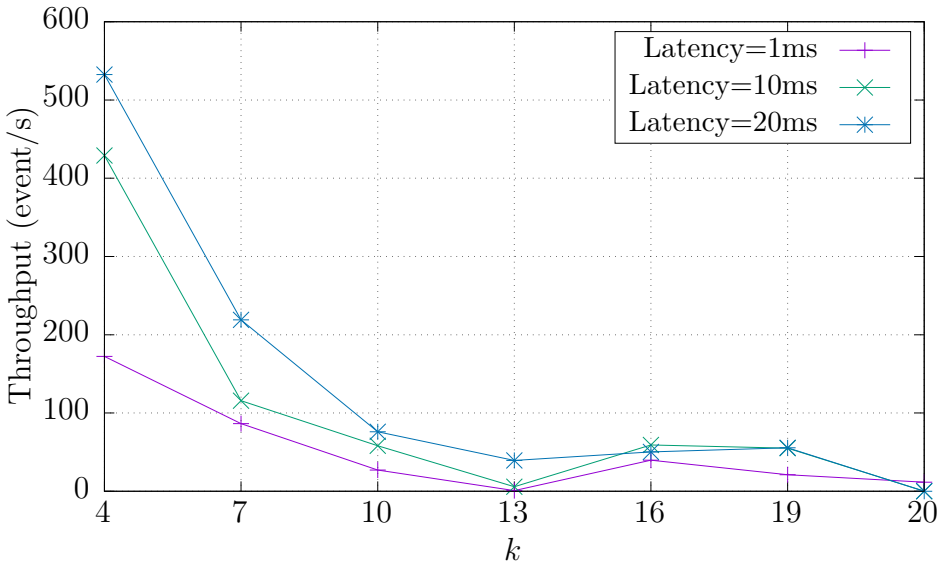

Figure 6.8: The impact of the network latency on the effectiveness of micro-transactions in throughput

becomes more important with the increasing *latency*. For instance, when $latency = 20ms$ and $k = 7$ results show that micro-transactions outperforms the full-replication protocol by ~ 219 event/s regarding throughput and $\sim 1.8s$ for the latency. It can be noticed from the above results that using micro-blocks

relatively enhances the overall performance of the proposed protocol. That optimized efficiency will undoubtedly broaden the spectrum of the use cases where the proposed protocol can be applied. Besides, it reduces resources cost since the consensus communications are restrained to a few nodes. That will become more important in a larger vehicular network.

6.3.3 Micro-transactions protocol

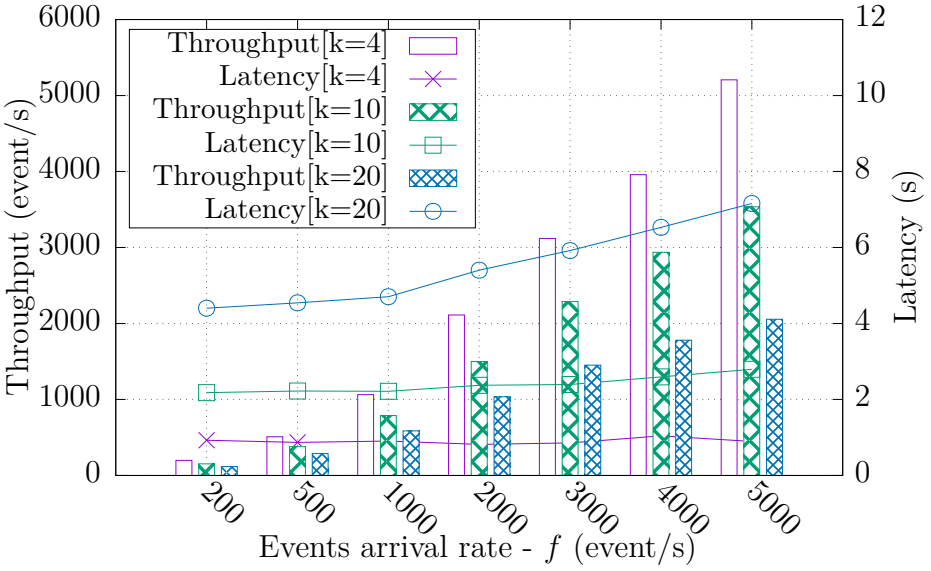

Figure 6.9: The impact of the events' arrival rate (f) and the replication factor (k) on the performance (i.e., latency and throughput).

Figure 6.9 illustrates the performance (i.e., throughput and latency) of the micro-transactions protocol for $k = 4$, $k = 10$, and $k = 20$ against the increasing rate of transaction generation (f). As expected, results show that the smaller the size of the consensus group (k), the better the performance. For example, for $k = 4$, the system throughput can reach ~ 5207 event/s with an average event confirmation time less than $900ms$. While in the worst case ($k = 20$), the system throughput does not exceed 2056 event/s, and the latency can reach $7s$, which may not be suitable for some VANET applications. Note that the smaller the k, the more vulnerable the system is; therefore, trade-offs between required security and excepted performance should be decided on.

6.3.4 Communication load and storage overhead

In Figure 6.10 and Figure 6.11, 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.

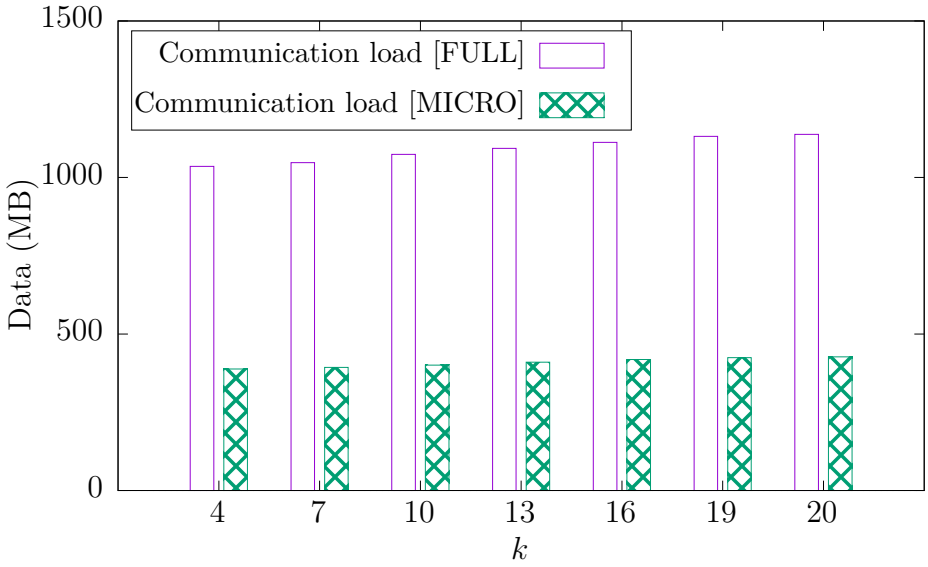

Figure 6.10: The communication load vs. consensus group size - k

Figure 6.10 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 6.11 measures the blockchain size with the increasing consensus partic-

ipants number (k). The plotted results represent the average size in MB of the local copy of the blockchain stored by the RSUs.

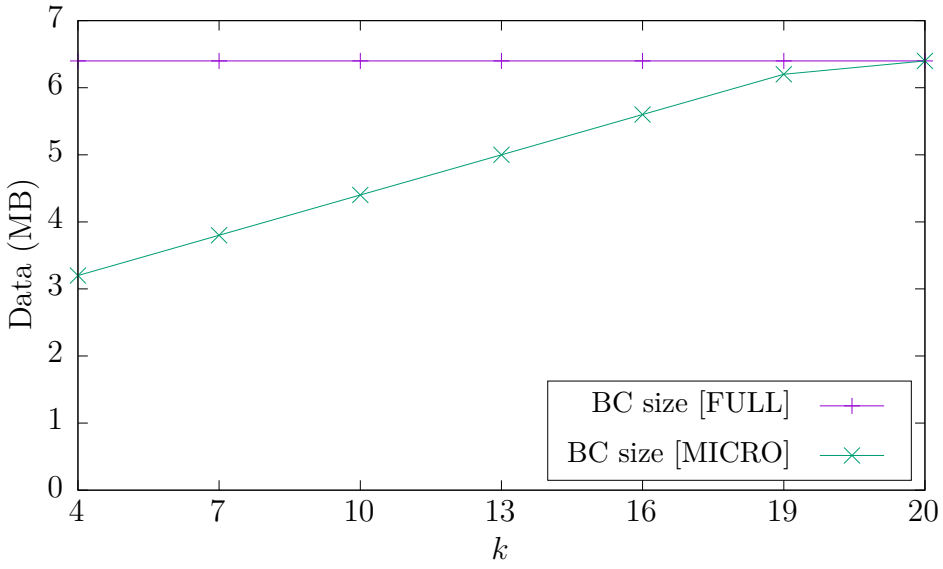

Figure 6.11: The storage cost vs. consensus group size - k

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 3.2 MB to 6.4 MB ($\times 2$). The results also show that if $k = 20$, the blockchain's size becomes equal to Full-replication. 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 blockchain's size (BC size) remains invariant because blocks are replicated across all RSUs. We wait until all generated transactions are finalized before measuring the size of the blockchain.

6.3.5 The k-Replication vs. the PoW-based protocol

In Figure 6.12, we depict the throughput and latency of the PoW-based protocol proposed in chapter 4 and the k-Replication protocol while varying the number of traffic events generated per second (f). We compare the performance of two configurations of these protocols, a PoW-based blockchain of a difficulty $d = 4$ and

a k-Replication protocol where $k = 7$. Both PoW (PBFT-PoW) and round-robin (PBFT-Robin) for the leader election are considered for the latter.

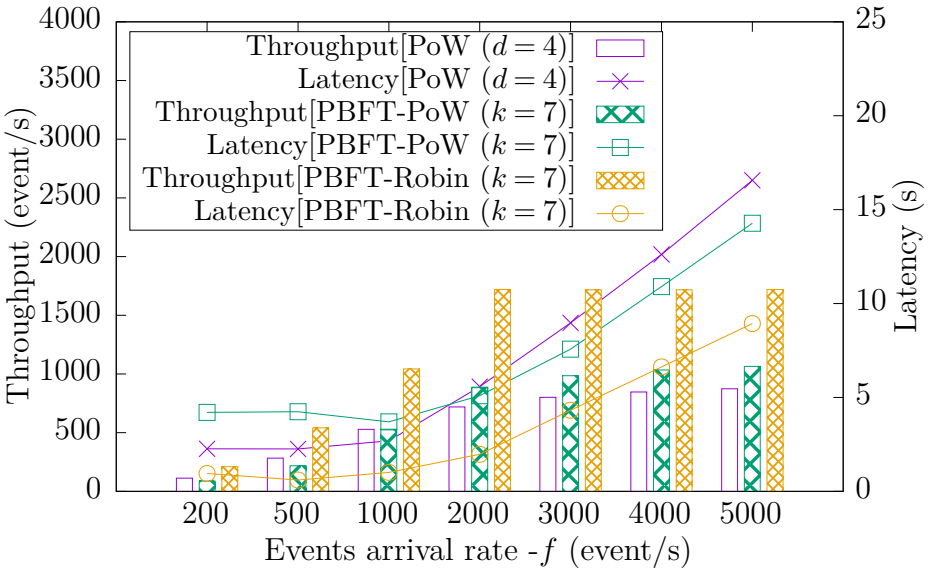

Figure 6.12: Comparison between the performance of PBFT and PoW

Figure 6.12 shows that using a Round-robin for leader election in the k-Replication protocol increases the number of traffic events that can be validated and added to the blockchain per second. However, this compromises the blockchain ability to withstand DDoS attacks against the leader. Therefore, it is essential to randomly select the leader, using a random mechanism that makes the next block proposer unknown, to minimize DDoS attacks on its resources.

Finally, as can be seen from Figure 6.12, the PBFT-based protocol has always better throughput and less latency in comparison with PoW. That is because PoW implies a significant delay in puzzle solving. However, from a security point of view, the PoW might be better than PBFT (k=7), which supports only 2 (7/3) Byzantine RSUs.

6.3.6 Comparison with existing works

To our knowledge, DSSCB [134] and paper [135] are the closest protocols to ours. We rely on the same consensus algorithm, PBFT, which is a proven correct-voting-based consensus algorithm [133]. Therefore, these two works have been selected for comparison with the proposed protocol.There are other related works that we have omitted due to the difference in the consensus mechanism and the security

guarantees (cf.3.4.5). For example, Kang et al. [129] introduced a protocol similar to the one proposed in this paper; nevertheless, the voting consensus model was different. They introduced a new voting-based consensus model that operates in four rounds of communications (one round more than PBFT). In addition, the consensus protocol relies heavily on the consensus leader to progress; this makes it more vulnerable to DDoS attacks on the leader. To summarize, the proposed consensus scheme needs to be positioned with respect of PBFT, mainly regarding security guarantees.

This section compares these schemes with the two configurations of the proposed scheme: Full-Replication and k-Replication (micro-transactions protocol). Table 6.2 illustrates this comparison while focusing on our key contributions: decentralization, latency, communication and storage costs, and road traffic event validation efficiency.

Protocols	**Decentralization**	**Latency**	**Communication and storage costs**	**Validation efficiency**
DSSCB [134]	+	10 minutes	+++	+
Paper [135]	+	-	+++	+
Full-Replication	+++	≤ 1 minute	+++	+++
k-Replication	++	≤ 1 minute	+	+++

Table 6.2: Comparative table between the proposed protocol and close approaches in the literature

- **Decentralization :** the robustness and reliability of blockchain protocols rely heavily on the decentralization property. This property ensures transparency and enhances the robustness of the system. For example, in DSSCB [134], centrally pre-selected RSUs orchestrate block validation, so the consensus RSUs are known in advance and thus make them more vulnerable to DoS attacks. Similarly, in [135] a PBFT-based protocol was adopted in which pre-selected RSUs (chosen by the Department of Transportation and consortium members) were selected. Like the previous protocol, this one has centralization issues, and the pre-selected RSUs will be the target of DoS attacks, which will affect the progress of the blockchain. We mitigate these centralization issues, in our work, by dynamically selecting the consensus RSUs for each new block. More importantly, unlike

in the mentioned protocols, consensus RSUs blocks are not designated by any central authority. We recall here that the central authority is relied on only for authentication credentials generation in our proposal. Furthermore, all RSUs can participate in the consensus, which allows for greater decentralization and network resilience. Finally, the k-Replication configuration of our protocol restricts block replications to consensus RSUs. Therefore, it is less decentralized because the non-consensus RSUs do not store the full block (only micro-blocks).

Regarding the number of supported faulty/malicious RSUs, all these protocols support the same number of malicious RSUs, which is the 33% of the consensus RSUs.

- **Latency :** in terms of latency, the proposed protocol outperforms both DSSCB and the proposed scheme in [135]. In the latter, the evaluated metrics are related to packets delivery ratio over RSUs, packets receiving rate, and MAC/PHY layer overhead. No performance metrics such as latency were provided. On the other hand, the DSSCB protocol has a transaction confirmation delay of 10 minutes. This delay will limit the adoption of such a protocol for road traffic events validation. We proposed a more scalable protocol by relying on Round-robin protocol instead of PoW when electing PBFT consensus leader. We also were able to reduce the latency further through micro-transactions in the k-Replication configuration.

- **Communication and storage costs :** these four protocols rely on PBFT. Thus, consensus RSUs must process 3 rounds of communication to reach an agreement on the next block to be added to the blockchain. This process implies significant communication overhead. Moreover, the storage requirements are consequential in the case where each RSU maintains the entire blockchain. In the k-Replication setup, we reduced both the communication and storage costs required to maintain the blockchain. This is done by relying on micro-transactions.

- **Validation efficiency :** traffic events validation is more complex than simply verifying signatures. Evaluating the plausibility of a traffic event requires the testimony of passing vehicles around the event location. With its detection devices, a vehicle can witness a relevant event and inform nearby RSUs. Therefore, the closer a RSU is to an event, the sooner it is reported of that event. For this reason, consensus established by a fixed pre-selected RSUs, as is the case in DSSCB [134] and paper [135], may require sharing event appearance evidence to these pre-selected RSUs. That will imply additional delay and communication load on the RSUs layer. Instead, we simplify this process by considering event appearance location so as event validation is

orchestrated by geographically closer k RSUs (see section 6.2.3). Thus, it is unnecessary to disseminate the evidence related to road traffic events to all RSUs.

6.4 Results Discussion

The above results showed that the k-Replication protocol, in it best configuration, can validate traffic event in less than 900 ms. This delay is good enough to efficiently advertise traffic announcements. Results have also indicated that the proposed protocol outperforms the adapted PoW protocol in the chapter 4. This is without compromising the decentralization property of the blockchain as it is the case in [134, 135], since in the k-Replication protocol all RSUs participate into the consensus protocol dynamically.

Furthermore, results have pointed out that besides minimizing the storage and communication cost of the overall system, micro-transactions enhance the performance of the blockchain, especially when the network latency is significant.

Finally, results showed that the performance of k-Replication decreases with the increasing consensus group size (k). k is directly related to the robustness and security of the protocol, so that the larger k is, the more the supported malicious RSUs. Therefore, minimizing the size of the consensus group leads to a trade-off between the system performance and security.

6.5 Conclusion

This chapter presented a new blockchain-based protocol to secure traffic messages in VANET. The RSUs are used as blockchain nodes; they receive traffic events from vehicles and validate their trustworthiness using a dynamic PBFT protocol. The proposed scheme dynamically selects RSUs to participate in the consensus. The selection is based on the proximity of RSUs from the traffic events themselves. Furthermore, micro-transactions which are compressed versions of the original traffic events, reduce communication and storage costs.

We evaluate different instances of the studied protocol and assess several metrics such as the throughput, latency, storage, and communication load. Finally, a performance comparison between the k-Replication protocol and the PoW-based protocol in chapter 4 is conducted; the obtained results have shown that the k-Replication protocol drastically reduces the event confirmation delay without relying on a pre-defined consensus group.

The k-Replication protocol achieves its best performance when the consensus group is minimized. However, in such a configuration, the number of supported malicious RSUs is reduced. It is interesting to keep the consensus group minimized without compromising the reliability of the consensus group. The next chapter proposes a trust and distance-based model to elect the most efficient consensus group considering event appearance location.

7 Trust model to scale and secure traffic events management

Contents

7.1 Introduction

This chapter proposes a trust model to improve the security of the proposed protocol in chapter 6. As a reminder, in chapter 6, we presented the k-Replication protocol, which is based on k RSUs elected dynamically according to their proximity to traffic events appearance. Results showed that k must be kept minimum to minimize traffic event confirmation latency. But that deteriorates the system security. Therefore, we propose to select the most reliable consensus participants

(validators) based on a trust model. This latter is validated by simulation, and its efficiency regarding decentralization and fairness of block-creation is assessed under various attacking scenarios.

The remainder of this remainder is organized as follows. Section 7.2 presents related works. Section 7.3 details the proposed trust model. Section 7.4 discusses the security model. Finally, section 7.5 concludes this chapter.

7.2 Related Works

Various trust models have been proposed in Vehicular Network (VN). Most of them are designed for vehicular trust management Siddiqui et al. [146]. For example, Minhas et al. [147] proposed a trust model that detects erroneous data generated by vehicles. In their proposed protocol, each vehicle maintains the trust of its neighbours, helping to assess received messages from them. But it is difficult to determine a vehicle's overall trust due to the high mobility. As a solution, Mármol and Pérez [148] relied on RSUs to evaluate vehicle trust and isolate malicious or selfish cars from spreading inaccurate information. That is an exciting scheme, as the vehicle's global trust is accessible at the RSUs level.

However, recent studies have highlighted the problem of vehicle trust storage, as neither vehicles nor RSUs are fully reliable. These studies use a blockchain to maintain a decentralized database of vehicle trust to address the challenge. For instance, Lu et al. [149] proposed a blockchain that records all the broadcasted messages by the vehicles. The goal is to enable persistent evidence to trace vehicle reputation. In Lu et al. [149], vehicles are the global consensus providers through PoW, which induces significant latency in updating vehicle trust. To alleviate that issue, Yang et al. [143] propose a RSUs-based blockchain of vehicle trust; they rely upon a combined PoW and PoS consensus mechanism. That reduces the consensus burden on vehicles by uploading it to the RSUs, which has more resources.

In this work, we don't manage vehicle trust. Instead, we focus on the trust of block validators, which are the RSUs. The goal is to leverage that trust to select the most appropriate validators for each block. Not only. The event appearance is also as well taken into consideration.

7.3 Trust Model

This section presents the proposed scheme: a trust model designed to improve the reliability of block validators without compromising the decentralization of

block-creation. It starts with the definition of RSU trust, in section 7.3.1. Then, section 7.3.2 elaborates the adopted methodology to select block validators. Finally, section 7.3.3 details the complete system workflow.

7.3.1 RSU trust

To choose a set of block validators, we define a trust score for each RSU. This score increases with correctly validated traffic events and decreases if a RSU fails to propose a valid block. RSU trust is updated based on the following rules:

Let c_i be the trust value of RSU i. And N the number of traffic events in the current block to be validated.

$$c_i \leftarrow c_i + \alpha_c * T \ ; \ T = \frac{N}{N_{max}} \text{ and } \alpha_c \in [0,1]$$

Where N_{max} is the maximum number of events allowed in one block, and α_c is the system parameter indicating the increasing speed of the trust value.

On the flip side, when a RSU fails to create a block, its trust decreases according to the following formula :

$$c_i \leftarrow c_i - \beta_c * T \ ; \ \beta_c \in [0,1]$$

Where β_c is the system parameter defining the decreasing speed of the trust value.

7.3.2 Validators selection

The trust is a crucial parameter to filter out faulty/malicious RSUs when selecting block validators. But also, the geographical distance between the validating RSUs and the event occurrence location must be considered. The aim is to form the most reliable validators group. Note, the closer a RSU is to an event appearance location, the higher its probability of having evidence related to that event. Therefore, in addition to RSUs trust, we consider their proximity to events. We define a score value associated to each block, considering both the RSUs trust and their proximity to event occurrence location. This score is calculated as follows:

First, let's consider block b the current block to be validated and the set $\{e_1, e_2, ..., e_N\}$, its composing road traffic events. The proximity of a given rsu_j to block b is computed by :

$$d_j^b = \frac{1}{N} \sum_{k=1}^{N} d(e_k, rsu_j)$$

Where $d(e_k, rsu_j)$ defines the Cartesian distance between the location of the event k and the rsu_j.

Let assume $C_j = \{r_k^c\}/k \neq i$ and $D_j^b = \{r_k^d\}/k \neq i$ are respectively the sorted list of RSUs according to their trust scores, and their proximity to block b. The index i corresponds to the block proposer, which is a validator by default. C_j is sorted ascending such as the RSU with the highest trust score have the highest rank. And D_j^b is sorted decreasing, such as the RSU with the lowest d_j^b is ranked last.

Next, to each potential validator $rsu_{j\neq i}$, we associate a couple $(r_j^c, r_j^d) \in (C_j, D_j^b)$ such that r_j^c defines the rank of the RSU j based on its trust, and r_j^d its rank based on its proximity to b. These two values are used to compute a score s_j corresponding to chances of $rsu_{j\neq i}$ to be selected as validator of the block b. s_j is expressed based on r_j^c and r_j^d as follows.

$$s_j = \alpha * r_j^c + \beta * r_j^d \ ; \ \alpha, \beta \in [0,1] \text{ and } \alpha + \beta = 1$$

Where α and β are the system parameters representing the weight of reputation and proximity to events, respectively.

Finally, we define $S_j = \frac{s_j}{\sum_{k'=1}^{N-1} s_{k'}}$ to normalize s_j in $[0,1]$. S_j is the probability of the rsu_j to validate the block b. The block validators are chosen based on this probability. As illustrated in Algorithm 5 the RSUs with the highest S_j are selected such as the cumulative sum of their S_j do not exceed a fixed threshold S_{th}. This latter represents the required security level in the system. For example, If $S_{th} = 1$, then all RSUs will validate all blocks, which opposes the main goal to minimize block validators. Consequently, S_{th} should be controlled as a trade-off between the validators size (k) and the blockchain security.

7.3.3 Scheme description

This section details how the trust model is integrated into the blockchain system. The following five steps outline the system's workflow from the selection of block proposer to the blockchain update.

Algorithm 5 Validators Selection

Input :
L: a collection of RSUs with their associated scores sorted decreasing
S_{th} : The threshold
Output :
v: list of elected validators

1: $v \leftarrow \{\}$ ▷ Initialisation
2: $S \leftarrow 0$
3: **repeat**
4: $tmp \leftarrow L.pop()$
5: $S \leftarrow S + tmp.getScore()$
6: $v.append(tmp.getRSU())$
7: **until** $S >= S_{th}$
8: **return** v

- **Step 1 :** a round-robin selection is used to determine the next block proposer. However, there are conditions based on the RSUs trust. To propose a block, a RSU trust score must be greater than a fixed parameter c_{min}. In addition, if a RSU fails to create a block, it will be skipped for the following b_{skip} blocks. c_{min} and b_{skip} are parameters of the trust model; they must be adjusted to minimize the risk of selecting a faulty/malicious RSU as block proposer. Note that the selection of a trusted RSU as a block proposer is a crucial step for successful block creation. Therefore, RSUs with low trust scores must gain more trust before becoming eligible for block proposing. That is possible since even with a trust score below c_{min}, a RSU is still eligible for block validation and thus can gain more trust. The goal is to filter out malicious RSUs from proposing blocks while ensuring the decentralization and fairness of the block proposition process.

  The RSUs trust scores are initialized to c_{min}. Furthermore, To avoid collisions, RSUs with the same trust scores are sorted based on their public identity.

- **Step 2 :** at this level, one RSU has the token to propose the next block in the blockchain. Thus, it creates a block of time-stamped road traffic events and then selects validators according to the methodology described in section 7.3.2.

- **Step 3 :** Once the block proposer has selected the validators, it is time to proceed with the consensus (PBFT). At the end of the consensus, the block will be validated and signed by each validator, including the block proposer.

Next, the block will be shared to the rest of the RSUs (non-validators) (cf. 6.2.3).

- **Step 4 :** This step consists of block verification once received by the non-validators. It includes checking the block format and then updating RSUs trust. As described in the algorithm 6, the block verification starts by checking the block proposer trust. If it is not greater than c_{min}, the block is rejected by the correct RSUs. Next, the block score S_b is calculated and must be less than the threshold S_{th}. Finally, the validators selection process is reverse-engineered to verify whether it was according to the trust model rules. If the result is negative, the block proposer is not rewarded. Otherwise, the validators, including the block proposer, are rewarded equally.

  It is possible that the block proposer is not responsible for not following validators selection rules. However, in this study, we do not address this scenario. Instead, we assume that malicious RSUs attack when they are block proposers; therefore, only the block author is punished if the trust model rules are not followed.

7.4 Security Model

The main objective of the proposed trust model is to reduce the number of RSUs involved in block validation; and that without compromising the decentralization or security of the blockchain. Reducing the number of block validators certainly improves the performance of the blockchain, as it takes less time to reach a consensus within a small group. However, on the other hand, the security of the blockchain will worsen. Therefore, the validators group should be as trustworthy as possible to cope with that.

The proposed solution chooses the most reliable RSUs (based on their contribution in the blockchain and their proximity to events location) to participate in the consensus. This reliability is expressed by a system parameter S_{th}, which represents the desired security level. For example, $S_{th} = 33\%$ corresponds to $1/3$ of the RSUs voting power. That means the third most appropriate RSUs validate blocks. Later, in the evaluation section, S_{th} is varied to study its impact on the trust model.

Algorithm 6 Block Verification based on Trust Model rules

Input:
i : block proposer index
b : Block
k : block validators size
Workflow:

Verify block format : hashes and signatures
Verify $c_i >= c_{min}$
$S_b \leftarrow \sum_j S_j$; $j \in b.getValidators()$
Verify that $S_b \leq S_{th}$
Compute sorted list $\{S_{j \neq i}\}$ of the RSU scores based on the block b
punish= FALSE
for each $j \in b.getValidators()$ **do**
    if rank(j,$\{S_{j \neq i}\}$) $\leq k-1$ **then**
        $c_j \leftarrow c_j + \alpha_c * T$ ▷ Reward validators
    else if not punish **then**
        punish= TRUE
    end if
end for
if punish **then**
    $c_i \leftarrow c_i - \beta_c * T$ ▷ Punish the block proposer
end if

7.5 Evaluation

To evaluate the proposed model, this section simulates various instances of the trust protocol. The evaluated parameters are fairness (i.e., all well-behaving RSUs have equal chances to propose the next block and are also equally rewarded), the decentralization of the proposed model, and its effectiveness in reducing block validators size (k) are measured. Note, k is directly linked to the performance of the blockchain. The lower it is, the better the performance.

The proposed trust model is implemented using a server with the following properties: Dell R640 server, Intel(R) Xeon(R) Silver 4112, 2.60 GHz CPU, 8-core CPU, 64 GB RAM, and running Ubuntu 18.04. Some simulator metrics are fixed to study others. For instance, α and β are set to 0.5. i.e., the RSUs trust and proximity of events have the same weight. In addition, α_c : the increase rate of RSU reward is set to 0.1 and β_c : the punishment coefficient to -1. Thus, it takes time to build solid trust, and any misbehaving is strongly punished. Moreover, T

and b_{skip} are set to 1, the number of RSUs is set 20, and c_{min} to 0.5.

On the other hand, the threshold S_{th} takes the values 33%, 50% and 66%. These latter, correspond to the 1/3, 1/2, and 2/3 RSUs with the highest global score (trust and distance combined). Table 7.1 summarizes the simulation parameters. Unless otherwise mentioned, $s_{th} = 33\%$.

The simulation mimics the proposed protocol from block proposal to blockchain update. We generate 1000 blocks while recording the average validators size (k) and RSUs trust scores after each block. This process is repeated 20× with different seeds, and then the means are plotted. For simplicity, we do not capture the exact event occurrence location. Instead, we randomly shuffle the RSUs ranks regarding block proximity.

Parameters	**Value(s)**
# of generated block	$[1:1000]$
# of RSUs	20
α	0.5
β	0.5
α_c	0.1
β_c	-1
T	1
b_{skip}	1
c_{min}	0.5
S_{th}	33%, 50%, 66%

Table 7.1: Trust model simulation parameters

7.5.1 Results

This section evaluates the effectiveness of the proposed trust model. We consider three scenarios. In Scenario 1, all RSUs are considered benign, i.e., all RSUs propose valid blocks. In scenario 2, 33% (6) of the RSUs are malicious; they propose incorrect blocks whenever they are elected as block proposers. Finally, scenario 3 considers a more complex attack in which 33% of RSUs randomly propose an invalid block when they are block proposers.

For each scenario, the effectiveness of the trust model in minimizing the number of validators (k) is evaluated. Similarly, its ability to isolate malicious RSUs is examined.

Scenario 1:

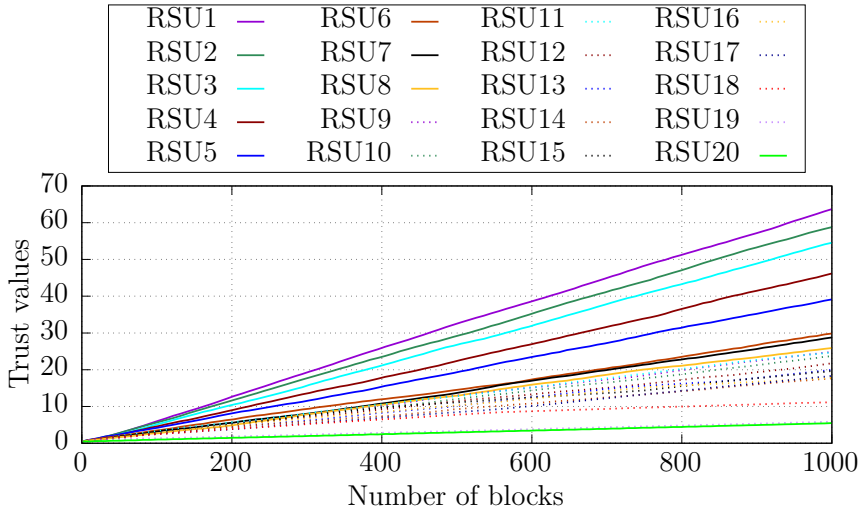

Figure 7.1: Evolution of RSUs trust scores (Scenario 1)

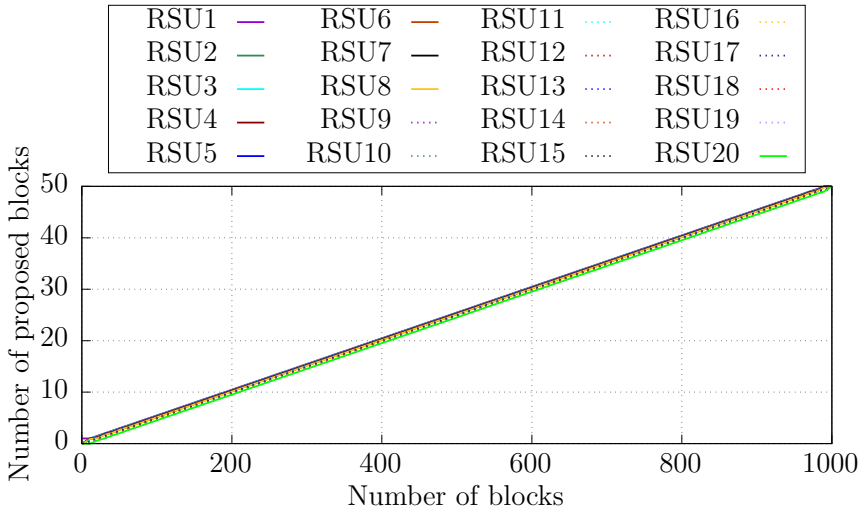

Figure 7.2: Number of proposed blocks

To evaluate the decentralization of the trust model, Figure 7.1 trace the evolution of the RSUs trust scores with the increasing number of blocks. The results show that all RSUs trust increase with time. That is because all RSUs are behaving correctly and thus getting rewarded for block validation.

Results also show that the RSU1 trust score is 47 units of trust higher than the RSU20 trust score. This is because the system initialization made the RSU1 the first block proposer. Also, besides being the most trusted among the RSUs since

the early stage, RSU1 had the chance to be well ranked related to the distance. On the other hand, the RSU20 trust is low because of similar reasons.

The fairness of the proposed model is evaluated in Figure 7.2; it is expressed by the number of proposed blocks by each RSU. The results show the fairness of the block-creation process. For example, when there are 1000 blocks in the blockchain, each RSU proposes 50 as shown in Figure 7.2. This is because all RSUs are correct and always eligible to create blocks.

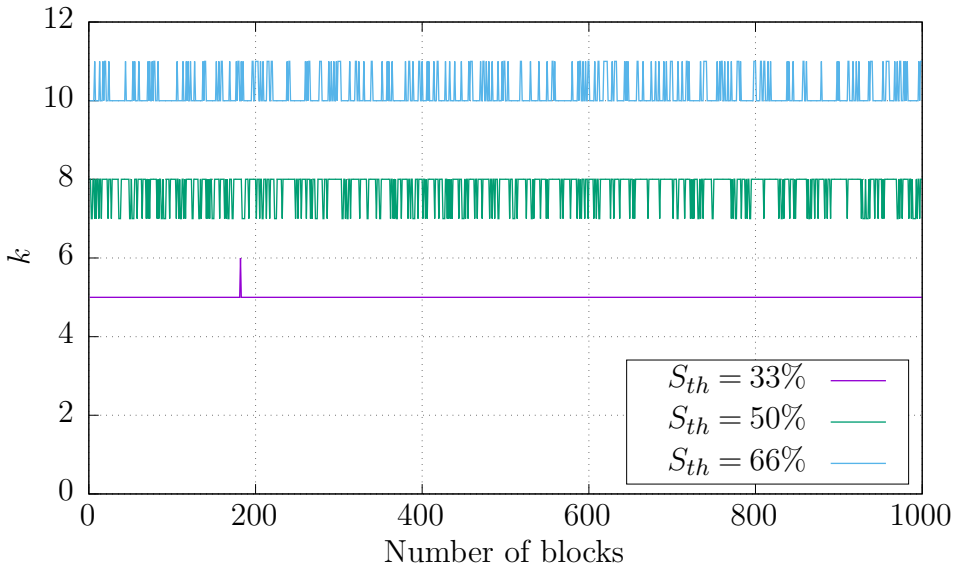

Figure 7.3: Block validators size evolution (Scenario 1)

Figure 7.3 evaluates the effectiveness of the trust model in minimizing the validators group. As can be seen in Figure 7.3, the validators group size (k) increases with increasing S_{th}. For example, when $S_{th} = 66\%$ the block validators are between 10 and 11, while if $S_{th} = 33\%$ only 5 to 6 validators are sufficient. In this last case, even if 1 RSU is malicious, the protocol is still secure, thanks to the PBFT. Note that minimizing k is crucial in improving blockchain global performance.

Finally, the proposed model can reduce the number of block validators by prioritizing the most reliable and suitable (trustworthiness and distance combined) RSUs.

Scenario 2 & 3

The above results showed the decentralization, fairness, and efficiency of the proposed model to minimize the size of block validators. However, malicious RSUs were not considered. This section focuses on evaluating the trust model in the

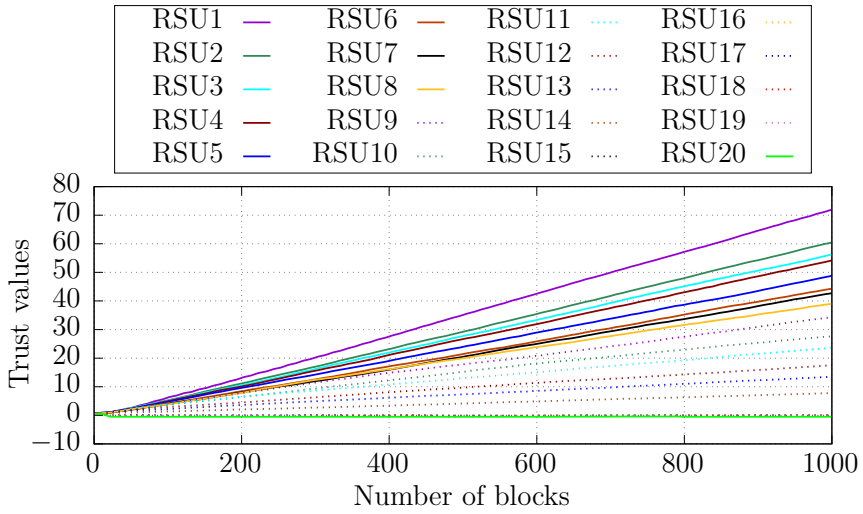

Figure 7.4: Evolution of RSUs trust (Scenario 2)

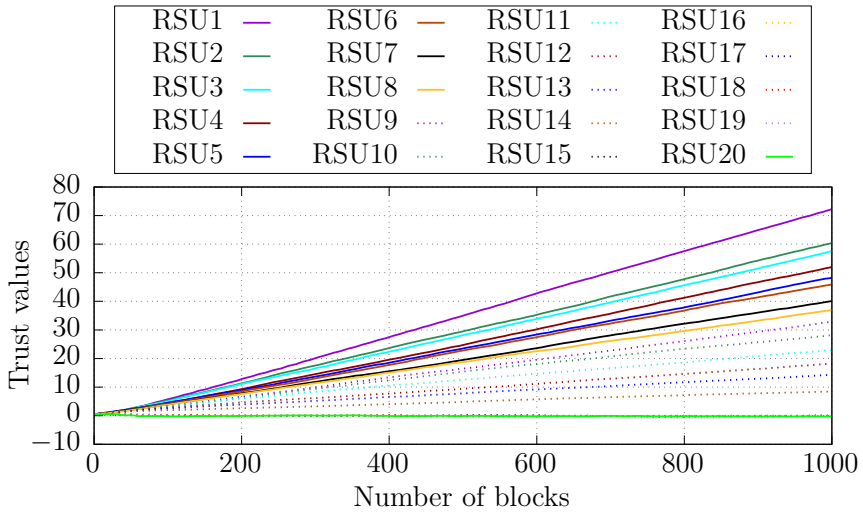

Figure 7.5: Evolution of RSUs trust (Scenario 3)

presence of attacks. We consider the two attack scenarios defined above (Scenario 2 and 3) and assess the same metrics as in Scenario 1. In the forthcoming graphs, RSU15,RSU16,..,RSU20 are considered malicious according to these two scenarios.

Figures 7.4 and 7.5 show the evolution of RSUs trust scores respectively in Scenario 2 and Scenario 3. As can be seen in both scenarios, the malicious RSUs have been isolated from block validation and proposition. That shows the efficiency of the proposed model in dealing with attacks.

Moreover, from the decentralization and fairness point of view, the results are

positive as the trust values of the correct RSUs continue to grow while the trust values of the attackers remain low, thus preventing them from harming the system.

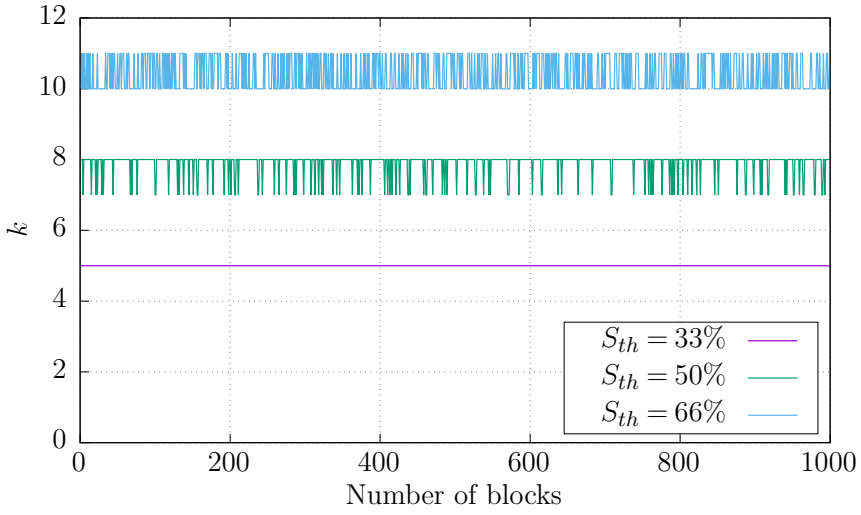

Figure 7.6: Block validators size (Scenario 2)

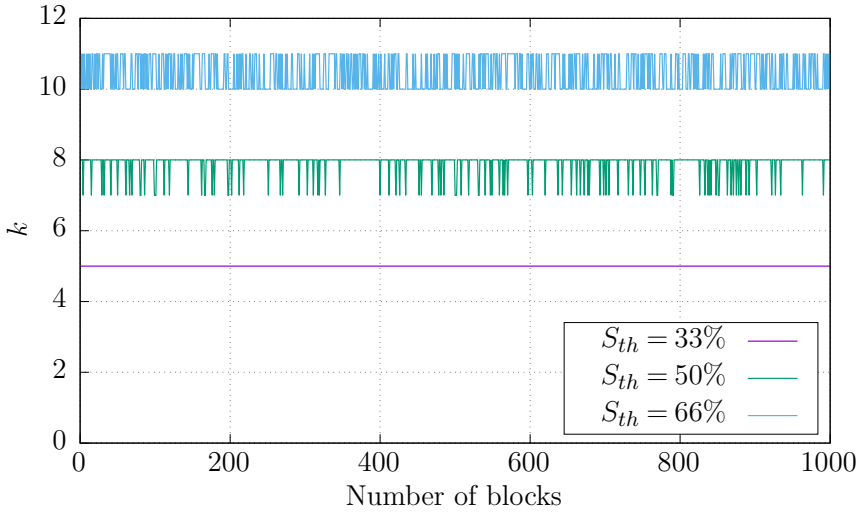

Figure 7.7: Block validators size (Scenario 3)

Figure 7.6 and Figure 7.7 show the average size of block validators (k) respectively in Scenario2 and 3. The results show that even in the presence of malicious RSUs, the proposed trust system successfully minimizes k. For example, in both scenarios 2 and 3, k is less than 5 if $S_{th} = 33\%$. Consequently, thanks to the trust model, the blockchain system can achieve good performance even in the presence of malicious RSUs.

From a security perspective, $S_{th} = 33\%$ means that the most appropriate 1/3 of

RSUs based on distance and trust level are required to validate a block. Depending on the expected security level, this threshold can be increased. For example, as show Figure 7.6 and Figure 7.7, increasing S_{th} from 33% to 50% increases k by 2 validators. Therefore, $S_{th} = 50\%$ could be preferred since it increases the reliability of the block-creation without affecting the performance as much (2 additional validators).

Moreover, the number of supported malicious RSUs by PBFT ($F = \frac{1}{3}(N-1)$, where N is the number of consensus participants) can be exploited to further minimize k. For example, $k = 5$ and $k = 6$ can be reduced to $k = 4$ and $k = 7$ and $k = 8$ to 7 whitout affecting the consensus security.

7.6 Conclusion

This chapter proposed a trust model that allows the most appropriate group of RSUs to validate road traffic events. The RSUs trust scores and the road traffic events locations were relied on to build this group of validators. The decentralization and efficiency of the proposed protocol were evaluated through simulation of various scenarios. The results showed the capacity of the proposed model in filtering out malicious RSUs while ensuring decentralization and fairness of block validation. Moreover, results showed that the block validators size was reduced while choosing the best validators group for each block. These two last points enhance the security and performance of the proposed protocol in chapter 6.

8 Conclusion and Perspectives

Contents

This chapter concludes the thesis. Section 8.1 summarizes the thesis contributions, and section 8.2 gives perspective for future works.

8.1 Conclusion

Advances in wireless and sensor technologies have enabled autonomous vehicles to detect and report traffic events (e.g., accidents, traffic state, attack reports, etc.). However, these reported data must be available to vehicles and transparently verifiable so that they can be exploited to improve the transportation system. This thesis addresses the obstacles to providing a decentralized, transparent, and attack-resistant system for road traffic data management. The proposed solutions are based on blockchain technology because of its exciting properties in enabling transparency and decentralization in an untrusted network like VANET.

In chapter 2, we laid the background knowledge on the blockchain technology, its properties and challenges. We also presented VANET architecture, characteristics, applications requirements and challenges.

Blockchain integration into VANET

In chapter 3, we conducted a literature review on existing VANET architectures and highlighted the limits of each one. We provided a taxonomy of existing traffic data management architectures. We started with centralized, passed by distributed, and ended with decentralized architectures. We shed light on the limitations of each architecture while giving a particular focus on blockchain integration into VANET. The review showed the immaturity of existing blockchain-enabled VANET architectures–the main impediments are adaptability, scalability, and performance evaluation. Furthermore, blockchain integration into VANET is none trivial; different approaches exist. We proposed to rely on the RSUs as the blockchain maintainers instead of the vehicles.

Blockchain simulator

Throughout the thesis, we have built a blockchain simulator to emulate VANET-enabled blockchain scenarios. The implemented simulator is built on NS-3, a network simulator. We were able to reproduce different VANET communications, and vehicle mobility was captured using real traces. The components of the proposed blockchain system were implemented and integrated as modules into NS-3. Our objective was to address the issue of performance metrics to validate the integration of blockchain into VANET.

PoW-based blockchain adaptation in VANET

In chapter 4, we proposed a Proof of Work (PoW)-based blockchain to build a secure and decentralized database of traffic events. Although traffic events validation is more complex compared to financial transaction, the proposed scheme inherits all essential properties of the blockchain. The proposed protocol is validated through simulation, considering a blockchain deployed for a whole city. The evaluated metrics were the system performance (the throughput and latency). The results showed good performance for traffic efficiency applications by reducing the PoW difficulty.

The impact of attacking vehicles on the blockchain

In chapter 5, we investigated the impact of vehicles disseminating wrong events on the data reliability and the security of the blockchain. We used a threshold-based

method for event validation, where sufficient cars need to witness the event before validation. The results showed that the reliability of the blockchain data depends on the event validation approach. They also showed that relaxing the threshold induces more forks, which affects the consistency and security of the blockchain.

Scalable blockchain-adapted for road traffic data management

To minimize event confirmation in the blockchain latency, in chapter 6, we replaced the consensus mechanism Proof of Work (PoW) with Practical Byzantine Fault Tolerance (PBFT), which is less decentralized. However, we ensured the system's decentralization by dynamically selecting traffic event validators based on their proximity to the traffic event occurrence location. In addition, we were able to control the replication of the blockchain through micro-transactions. Micro-transactions minimize communication and storage cost of the blockchain. Finally, the proposed protocol has been deeply evaluated.

Trust model to isolate malicious validators

In chapter 7, we proposed a trust model to enhance the reliability of the traffic event validators, which are the RSUs. The proposed trust model is based on RSUs proximity to traffic events location and the RSUs reputation. The proposed approach was validated by simulation. The results have shown its decentralization and efficiency in minimizing the validators group without compromising the system security.

8.2 Perspectives

The thesis' contributions answer many questions towards blockchain inclusion in VANET. I believe that the proposed approaches advance towards more secure VANET applications and a better transportation system. However, the presented solutions could further be improved. We suggest a few improvements as follows:

- Scaling the blockchain for a wide region, such as a whole country. Clustering techniques, as in these studies [57, 58], cloud be explored. The challenge will be to ensure synchronization between different clusters in a reasonable delay.
- Introducing vehicles trust system to counter malicious vehicles from affecting the blockchain's data. There could be a trust-based blockchain coexisting

alongside the traffic data blockchain. In that way, the trust will be included at each layer of the Vehicular Network (VN).

- Prioritizing road traffic events based on their criticality to minimize road safety events' validation delay. In addition, the blockchain could be parameterized so that each class of events has its validation requirement.
- Investigating higher throughput distributed ledger. For example, DAG-based blockchains [150] could improve the processing of event validation and thus broaden the application area of the proposed approaches.

Author's Publications

Journals

1. El-Hacen Diallo, Omar Dib, and Khaldoun Al Agha. A scalable blockchain-based scheme for traffic-related data sharing in vanets. In *Blockchain: Research and Applications Journal*. Elsevier, **accepted**

Conferences

2. El-Hacen Diallo, Omar Dib, and Khaldoun Al Agha. A blockchain-based approach to track traffic messages in vehicular networks. In *Academia-Industry Consortium for Data Science (AICDS) in collaboration with Springer*, pages 1–5. Springer AISC, 2022

3. El-Hacen Diallo, Omar Dib, and Khaldoun Al Agha. The journey of blockchain inclusion in vehicular networks: A taxonomy. In *2021 Third International Conference on Blockchain Computing and Applications (BCCA)*, pages 135–142. IEEE, 2021

4. El-hacen Diallo, Omar Dib, Nicola Roberto Zema, and Khaldoun Al Agha. When proof-of-work (pow) based blockchain meets vanet environments. In *2021 12th International Conference on Information and Communication Systems (ICICS)*, pages 336–343, 2021

5. El-Hacen Diallo, Omar Dib, and Khaldoun Al Agha. An improved pbft-based consensus for securing traffic messages in vanets. In *The Twelfth International Conference on Information and Communication Systems (ICICS)*, pages 1–8. IEEE, 2021

6. El-hacen Diallo, Omar Dib, and Khaldoun Al Agha. On the adaptation of bitcoin-like blockchains for the secure storage of traffic-related events. In *International Conference on Advanced Information Networking and Applications*, pages 195–207. Springer, 2021

7. El-Hacen Diallo, Khaldoun Al Agha, Omar Dib, Alexandre Laube, and Hafedh Mohamed-Babou. Toward scalable blockchain for data management in vanets. In *Workshops of the International Conference on Advanced Information Networking and Applications*, pages 233–244. Springer, 2020

8. El-Hacen Diallo, Alexandre Laube, Khaldoun Al Agha, and Steven Martin. Efficient block replication to optimize the blockchain resources. In *2019 3rd Cyber Security in Networking Conference (CSNet)*, pages 1–5. IEEE, 2019

Book chapter

9. El-Hacen Diallo, Omar Dib, and Khaldoun Al Agha. Secure and scalable blockchain-enabled traffic data management using a trust model. In *Principles and Practice of Blockchains*, pages 1–8. Springer, **submitted**

Bibliography

[1] Grand View Research. Intelligent transportation system market growth & trends, . URL https://www.grandviewresearch.com/press-release/global-intelligent-transportation-systems-its-market.

[2] World Health Organization. Road traffic injuries. URL https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[3] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. Data fusion in intelligent transportation systems: Progress and challenges – a survey. *Information Fusion*, 12(1):4–10, 2011. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2010.06.001. URL https://www.sciencedirect.com/science/article/pii/S1566253510000643. Special Issue on Intelligent Transportation Systems.

[4] Xin Yu and Panos D Prevedouros. Performance and challenges in utilizing non-intrusive sensors for traffic data collection. 2013.

[5] Moumena Chaqfeh, Abderrahmane Lakas, and Imad Jawhar. A survey on data dissemination in vehicular ad hoc networks. *Vehicular Communications*, 1(4):214–225, 2014. ISSN 2214-2096. doi: https://doi.org/10.1016/j.vehcom.2014.09.001. URL https://www.sciencedirect.com/science/article/pii/S2214209614000448.

[6] Felipe Domingos Da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio AF Loureiro. *Data communication in VANETs: a survey, challenges and applications*. PhD thesis, INRIA Saclay; INRIA, 2014.

[7] Teodora Mecheva and Nikolay Kakanakov. Cybersecurity in intelligent transportation systems. *Computers*, 9(4):83, 2020.

[8] Euisin Lee, Eun-Kyu Lee, Mario Gerla, and Soon Y Oh. Vehicular cloud networking: architecture and design principles. *IEEE Communications Magazine*, 52(2):148–155, 2014.

[9] Kuljeet Kaur, Sahil Garg, Gagangeet Singh Aujla, Neeraj Kumar, Joel JPC Rodrigues, and Mohsen Guizani. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. *IEEE communications magazine*, 56(2):44–51, 2018.

[10] Ahmad Alhilal, Tristan Braud, and Pan Hui. Distributed vehicular computing at the dawn of 5g: A survey. *arXiv preprint arXiv:2001.07077*, 2020.

[11] Shubhanjali Sharma, Garima Gupta, and PR Laxmi. A survey on cloud security issues and techniques. *arXiv preprint arXiv:1403.5627*, 2014.

[12] Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park. A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, 75:200–222, 2016.

[13] Jianli Pan and James McElhannon. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, 5(1): 439–449, 2017.

[14] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing: A survey. *Future Generation Computer Systems*, 97:219–235, 2019.

[15] Jameela Al-Jaroodi and Nader Mohamed. Blockchain in industries: A survey. *IEEE Access*, 7:36501, 2019.

[16] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018.

[17] Vittorio Astarita, Vincenzo Pasquale Giofrè, Giovanni Mirabelli, and Vittorio Solina. A review of blockchain-based systems in transportation. *Information*, 11(1):21, 2020.

[18] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[19] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[20] Stephan Eichler. Performance evaluation of the ieee 802.11 p wave communication standard. In *2007 IEEE 66th Vehicular Technology Conference*, pages 2199–2203. IEEE, 2007.

[21] Hamssa Hasrouny, Abed Ellatif Samhat, Carole Bassil, and Anis Laouiti. Vanet security challenges and solutions: A survey. *Vehicular Communications*, 7:7–20 (9), 2017.

[22] Rashmi Mishra, Akhilesh Singh, and Rakesh Kumar. Vanet security: Issues, challenges and solutions. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 1050–1055. IEEE, 2016.

[23] Irshad Ahmed Sumra, Iftikhar Ahmad, Halabi Hasbullah, et al. Classes of attacks in vanet. In *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pages 1–5. IEEE, 2011.

[24] Lina Bariah, Dina Shehada, Ehab Salahat, and Chan Yeob Yeun. Recent advances in vanet security: a survey. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–7. IEEE, 2015.

[25] Mohammed Ali Hezam Al Junaid, AA Syed, Mohd Nazri Mohd Warip, Ku Nurul Fazira Ku Azir, and Nurul Hidayah Romli. Classification of security attacks in vanet: A review of requirements and perspectives. In *MATEC Web of Conferences*, volume 150, page 06038. EDP Sciences, 2018.

[26] Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

[27] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1:22–23, 2013.

[28] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2 (6-10):71, 2016.

[29] Grand View Research. Blockchain technology market size, share & trends analysis report, . URL https://www.grandviewresearch.com/industry-analysis/blockchain-technology-market.

[30] Yang Lu. The blockchain: State-of-the-art and research challenges. *Journal of Industrial Information Integration*, 15:80–90 (81), 2019.

[31] John Turek and Dennis Shasha. The many faces of consensus in distributed systems. *Computer*, 25(6):8–17 (9), 1992.

[32] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[33] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[34] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.

[35] Leslie Lamport. Fast paxos. *Distributed Computing*, 19(2):79–103, 2006.

[36] J-P Martin and Lorenzo Alvisi. Fast byzantine consensus. *IEEE Transactions on Dependable and Secure Computing*, 3(3):202–215, 2006.

[37] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong. Zyzzyva: speculative byzantine fault tolerance. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 45–58, 2007.

[38] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.

[39] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*, pages 436–454. Springer, 2014.

[40] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2567–2572 (2568). IEEE, 2017.

[41] Wenbing Zhao, Shunkun Yang, Xiong Luo, and Jiong Zhou. On peercoin proof of stake for blockchain consensus. In *2021 The 3rd International Conference on Blockchain Technology*, pages 129–134, 2021.

[42] Cong T Nguyen, Dinh Thai Hoang, Diep N Nguyen, Dusit Niyato, Huynh Tuong Nguyen, and Eryk Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. *IEEE Access*, 7:85727–85745 (85732), 2019.

[43] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). In Paul Spirakis and Philippas Tsigas, editors, *Stabilization, Safety, and Security of Distributed Systems*, pages 282–297, Cham, 2017. Springer International Publishing. ISBN 978-3-319-69084-1.

[44] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282–297. Springer, 2017.

[45] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

[46] Parinya Ekparinya, Vincent Gramoli, and Guillaume Jourjon. The attack of the clones against proof-of-authority. *arXiv preprint arXiv:1902.10244*, 2019.

[47] Peter Todd. Ripple protocol consensus algorithm review. *May 11th*, 2015.

[48] David Mazieres. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation*, 32, 2015.

[49] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, page 30. ACM, 2018.

[50] Arati Baliga, I Subhod, Pandurang Kamat, and Siddhartha Chatterjee. Performance evaluation of the quorum blockchain platform. *arXiv preprint arXiv:1809.03421*, 2018.

[51] Mike Hearn and Richard Gendal Brown. Corda: A distributed ledger. *Corda Technical White Paper*, 2016, 2016.

[52] Asad Ali Siyal, Aisha Zahid Junejo, Muhammad Zawish, Kainat Ahmed, Aiman Khalil, and Georgia Soursou. Applications of blockchain technology in medicine and healthcare: Challenges and future perspectives. *Cryptography*, 3(1):3, 2019.

[53] Umesh Bodkhe, Sudeep Tanwar, Karan Parekh, Pimal Khanpara, Sudhanshu Tyagi, Neeraj Kumar, and Mamoun Alazab. Blockchain for industry 4.0: A comprehensive review. *IEEE Access*, 8:79764–79800, 2020.

[54] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, 6(5):8076–8094 (page,1), 2019.

[55] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[56] Team Rocket. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies, 2018.

[57] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 931–948, 2018.

[58] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30. ACM, 2016.

[59] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[60] Serguei Popov. The tangle. *White paper*, 1(3), 2018.

[61] P Jovanovic. Byzcoin: Securely scaling blockchains. *Hacking, Distributed, August*, 2016.

[62] Ethan Buchman. Tendermint: Byzantine fault tolerance in the age of blockchains. Master's thesis, University of Guelph, 2016.

[63] Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, 2019. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2019.02.001. URL https://www.sciencedirect.com/science/article/pii/S030439751930091X. In memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part I.

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

[65] RVD Velde, MD Vries, AD Boer, A Das, and A Narain. Self-driving vehicles (sdvs) & geo-information, 2017. URL https://www.geonovum.nl/uploads/documents/Self-DrivingVehiclesReport.pdf.

[66] Brian Hayes. Cloud computing, 2008.

[67] Anthony D JoSEP, RAnDy KAtz, AnDy KonWinSKi, LEE Gunho, DAViD PAttERSon, and ARiEL RABKin. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[68] Mario Gerla. Vehicular cloud computing. In *2012 The 11th annual mediterranean ad hoc networking workshop (Med-Hoc-Net)*, pages 152–155. IEEE, 2012.

[69] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer applications*, 40:325–344, 2014.

[70] Rasheed Hussain, Zeinab Rezaeifar, and Heekuck Oh. A paradigm shift from vehicular ad hoc networks to vanet-based clouds. *Wireless Personal Communications*, 83(2):1131–1158, 2015.

[71] Ahmed Aliyu, Abdul Hanan Abdullah, Omprakash Kaiwartya, Yue Cao, Mohammed Joda Usman, Sushil Kumar, DK Lobiyal, and Ram Shringar Raw. Cloud computing in vanets: architecture, taxonomy, and challenges. *IETE Technical Review*, 35(5):523–547, 2018.

[72] Shadi Ibrahim, Bingsheng He, and Hai Jin. Towards pay-as-you-consume cloud computing. In *2011 IEEE International Conference on Services Computing*, pages 370–377. IEEE, 2011.

[73] Shanjiang Tang, Bu-Sung Lee, and Bingsheng He. Towards economic fairness for big data processing in pay-as-you-go cloud computing. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 638–643. IEEE, 2014.

[74] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.

[75] Rasheed Hussain, Junggab Son, Hasoo Eun, Sangjin Kim, and Heekuck Oh. Rethinking vehicular communications: Merging vanet with cloud computing. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pages 606–609. IEEE, 2012.

[76] Hyundai Motor Group | TECH. Connected car service. URL https://tech.hyundaimotorgroup.com/mobility-service/connected-car-service/.

[77] HYUNDAI AutoEver. Cloud platform. URL http://www.hyundai-autoever.com/common/goPage.view?page=m/en/service/digital_cloud.

[78] Rasheed Hussain, Fizza Abbas, Junggab Son, Donghyun Kim, Sangjin Kim, and Heekuck Oh. Vehicle witnesses as a service: Leveraging vehicles as witnesses on the road in vanet clouds. In *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, volume 1, pages 439–444. IEEE, 2013.

[79] Salim Bitam and Abdelhamid Mellouk. Cloud computing-based message dissemination protocol for vehicular ad hoc networks. In *international conference on wired/wireless internet communication*, pages 32–45. Springer, 2015.

[80] Stephan Olariu. A survey of vehicular cloud research: Trends, applications and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2648–2663, 2019.

[81] Kayhan Zrar Ghafoor, Marwan Aziz Mohammed, Kamalrulnizam Abu Bakar, Ali Safa Sadiq, and Jaime Lloret. Vehicular cloud computing: trends and challenges. In *Mobile Networks and Cloud Computing Convergence for Progressive Services and Applications*, pages 262–274. IGI Global, 2014.

[82] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344 (336–340), 2014. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2013.08.004. URL https://www.sciencedirect.com/science/article/pii/S1084804513001793.

[83] Rasheed Hussain, Zeinab Rezaeifar, Yong-Hwan Lee, and Heekuck Oh. Secure and privacy-aware traffic information as a service in vanet-based clouds. *Pervasive and Mobile Computing*, 24:194–209, 2015.

[84] Jun Zhou, Xiaolei Dong, Zhenfu Cao, and Athanasios V Vasilakos. Secure and privacy preserving protocol for cloud-based vehicular dtns. *IEEE Transactions on Information Forensics and Security*, 10(6):1299–1314, 2015.

[85] Lei Liu, Chen Chen, Qingqi Pei, Sabita Maharjan, and Yan Zhang. Vehicular edge computing and networking: A survey. *Mobile Networks and Applications*, pages 1–24, 2020.

[86] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2017.

[87] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78:680–698, 2018.

[88] Hang Liu, Fahima Eldarrat, Hanen Alqahtani, Alex Reznik, Xavier De Foy, and Yanyong Zhang. Mobile edge cloud system: Architectures, challenges, and approaches. *IEEE Systems Journal*, 12(3):2495–2508, 2017.

[89] Dario Sabella, Alessandro Vaillant, Pekka Kuure, Uwe Rauschenbach, and Fabio Giust. Mobile-edge computing architecture: The role of mec in the internet of things. *IEEE Consumer Electronics Magazine*, 5(4):84–91, 2016.

[90] Salman Raza, Shangguang Wang, Manzoor Ahmed, and Muhammad Rizwan Anwar. A survey on vehicular edge computing: architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing*, 2019, 2019.

[91] Yongxuan Lai, Hailin Lin, Fan Yang, and Tian Wang. Efficient data request answering in vehicular ad-hoc networks based on fog nodes and filters. *Future Generation Computer Systems*, 93:130–142, 2019.

[92] Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. Vehicular micro clouds as virtual edge servers for efficient data collection. In *proceedings of the 2nd ACM international workshop on smart, autonomous, and connected vehicular systems and services*, pages 31–35, 2017.

[93] Yongxuan Lai, Fan Yang, Jinsong Su, Qifeng Zhou, Tian Wang, Lu Zhang, and Yifan Xu. Fog-based two-phase event monitoring and data gathering in vehicular sensor networks. *Sensors*, 18(1):82, 2018.

[94] Tasneem SJ Darwish and Kamalrulnizam Abu Bakar. Fog based intelligent transportation big data analytics in the internet of vehicles environment: motivations, architecture, challenges, and critical issues. *IEEE Access*, 6: 15679–15701, 2018.

[95] Deepak Gangadharan, Oleg Sokolsky, Insup Lee, BaekGyu Kim, Chung-Wei Lin, and Shinichi Shiraishi. Bandwidth optimal data/service delivery for connected vehicles via edges. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 106–113. IEEE, 2018.

[96] Shumayla Yaqoob, Ata Ullah, Muhammad Akbar, Muhammad Imran, and Mohsen Guizani. Fog-assisted congestion avoidance scheme for internet of vehicles. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 618–622. IEEE, 2018.

[97] Xiao Chen and Liangmin Wang. Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—pepa. *IEEE Communications Letters*, 21(4):745–748, 2017.

[98] Zhenyun Zhou, Houjian Yu, Chen Xu, Zheng Chang, Shahid Mumtaz, and Jonathan Rodriguez. Begin: Big data enabled energy-efficient vehicular edge computing. *IEEE Communications Magazine*, 56(12):82–89, 2018.

[99] Wenyu Zhang, Zhenjiang Zhang, and Han-Chieh Chao. Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management. *IEEE Communications Magazine*, 55(12):60–67, 2017.

[100] Venkatraman Balasubramanian, Safa Otoum, Moayad Aloqaily, Ismaeel Al Ridhawi, and Yaser Jararweh. Low-latency vehicular edge: A vehicular infrastructure model for 5g. *Simulation Modelling Practice and Theory*, 98:101968, 2020.

[101] Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, and Juan Felipe Botero Vega. Security in sdn: A comprehensive survey. *Journal of Network and Computer Applications*, 159:102595, 2020.

[102] Shih-Chun Lin, Kwang-Cheng Chen, and Ali Karimoddini. Sd-vec: Software-defined vehicular edge computing with ultra-low latency. *arXiv preprint arXiv:2103.14225*, 2021.

[103] Alla Abbas Khadir and Seyed Amin Hosseini Seno. Sdn-based offloading policy to reduce the delay in fog-vehicular networks. *Peer-to-Peer Networking and Applications*, 14(3):1261–1275, 2021.

[104] Yueyue Dai, Du Xu, Sabita Maharjan, Guanhua Qiao, and Yan Zhang. Artificial intelligence empowered edge computing and caching for internet of vehicles. *IEEE Wireless Communications*, 26(3):12–18, 2019.

[105] Jie Guo, Bin Song, Yuhao Chi, Lahiru Jayasinghe, Chau Yuen, Yong Liang Guan, Xiaojiang Du, and Mohsen Guizani. Deep neural network-aided gaussian message passing detection for ultra-reliable low-latency communications. *Future Generation Computer Systems*, 95:629–638, 2019.

[106] Mashael Khayyat, Ibrahim A Elgendy, Ammar Muthanna, Abdullah S Alshahrani, Soltan Alharbi, and Andrey Koucheryavy. Advanced deep learning-based computational offloading for multilevel vehicular edge-cloud computing networks. *IEEE Access*, 8:137052–137062, 2020.

[107] Leo Mendiboure, Mohamed-Aymen Chalouf, and Francine Krief. Edge computing based applications in vehicular environments: Comparative study and main issues. *Journal of Computer Science and Technology*, 34(4):869–886, 2019.

[108] Hongwei Li, Rongxing Lu, Jelena Misic, and Mohamed Mahmoud. Security and privacy of connected vehicular cloud computing. *IEEE Network*, 32(3): 4–6, 2018.

[109] Xumin Huang, Rong Yu, Jiawen Kang, and Yan Zhang. Distributed reputation management for secure and efficient vehicular edge computing and networks. *IEEE Access*, 5:25408–25420, 2017.

[110] Hesham El-Sayed, Moumena Chaqfeh, Hadeel El-Kassabi, Mohamed Adel Serhani, and Henry Alexander. Trust enforcement in vehicular networks: challenges and opportunities. *IET Wireless Sensor Systems*, 9(5):237–246, 2019.

[111] Hesham El-Sayed, Sherali Zeadally, Manzoor Khan, and Henry Alexander. Edge-centric trust management in vehicular networks. *Microprocessors and Microsystems*, 84:104271, 2021.

[112] Leo Mendiboure, Mohamed Aymen Chalouf, and Francine Krief. Survey on blockchain-based applications in internet of vehicles. *Computers & Electrical Engineering*, 84:106646, 2020.

[113] Branka Mikavica and Aleksandra Kostić-Ljubisavljević. Blockchain-based solutions for security, privacy, and trust management in vehicular networks: a survey. *The Journal of Supercomputing*, pages 1–56, 2021.

[114] Lylia Alouache, Nga Nguyen, Makhlouf Aliouat, and Rachid Chelouah. Credit based incentive approach for v2v cooperation in vehicular cloud computing. In *International Conference on Internet of Vehicles*, pages 92–105. Springer, 2018.

[115] Madhusudan Singh and Shiho Kim. Blockchain based intelligent vehicle data sharing framework. *arXiv preprint arXiv:1708.09721*, 2017.

[116] Rakesh Shrestha, Rojeena Bajracharya, Anish P Shrestha, and Seung Yeob Nam. A new type of blockchain for secure message exchange in vanet. *Digital communications and networks*, 6(2):177–186, 2020.

[117] Matthew Wagner and Bruce McMillin. Cyber-physical transactions: A method for securing vanets with blockchains. In *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 64–73. IEEE, 2018.

[118] Muhammd Awais Hassan, Ume Habiba, Usman Ghani, and Muhmmad Shoaib. A secure message-passing framework for inter-vehicular communication using blockchain. *International Journal of Distributed Sensor Networks*, 15(2):1550147719829677, 2019.

[119] Seungmo Kim. Impacts of mobility on performance of blockchain in vanet. *IEEE Access*, 7:68646–68655, 2019.

[120] Vasiliy Elagin, Anastasia Spirkina, Mikhail Buinevich, and Andrei Vladyko. Technological aspects of blockchain application for vehicle-to-network. *Information*, 11(10):465, 2020.

[121] Ahmad Mostafa. Vanet blockchain: A general framework for detecting malicious vehicles. *J. Commun*, 14(5):356–362, 2019.

[122] JD Bruce. The mini-blockchain scheme. *White paper*, 2014.

[123] Yao-Tsung Yang, Li-Der Chou, Chia-Wei Tseng, Fan-Hsun Tseng, and Chien-Chang Liu. Blockchain-based traffic event validation and trust verification for vanets. *IEEE Access*, 7:30868–30877, 2019.

[124] Noureddine Lasla, Mohamed Younis, Wassim Znaidi, and Dhafer Ben Arbia. Efficient distributed admission and revocation using blockchain for cooperative its. In *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*, pages 1–5. IEEE, 2018.

[125] Rens W van der Heijden, Felix Engelmann, David Mödinger, Franziska Schönig, and Frank Kargl. Blackchain: Scalability for resource-constrained accountable vehicle-to-x communication. In *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, pages 1–5, 2017.

[126] Anderson Queiroz, Eduardo Oliveira, Maria Barbosa, and Kelvin Dias. A survey on blockchain and edge computing applied to the internet of vehicles. In *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2020.

[127] Rajesh Gupta, Aparna Kumari, and Sudeep Tanwar. A taxonomy of blockchain envisioned edge-as-a-connected autonomous vehicles. *Transactions on Emerging Telecommunications Technologies*, 32(6):e4009, 2021.

[128] Meng Li, Liehuang Zhu, and Xiaodong Lin. Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing. *IEEE Internet of Things Journal*, 6(3):4573–4584, 2018.

[129] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3):4660–4670, 2019. doi: 10.1109/JIOT.2018.2875542.

[130] Y. Yang, L. Chou, C. Tseng, F. Tseng, and C. Liu. Blockchain-based traffic event validation and trust verification for vanets. *IEEE Access*, 7:30868–30877, 2019.

[131] Lei Zhang, Mingxing Luo, Jiangtao Li, Man Ho Au, Kim-Kwang Raymond Choo, Tong Chen, and Shengwei Tian. Blockchain based secure data sharing system for internet of vehicles: A position paper. *Vehicular Communications*, 16:85–93, 2019.

[132] Benjamin Leiding, Parisa Memarmoshrefi, and Dieter Hogrefe. Self-managed and blockchain-based vehicular ad-hoc networks. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 137–140, 2016.

[133] Rachid Guerraoui, Nikola Knežević, Vivien Quéma, and Marko Vukolić. The next 700 bft protocols. In *Proceedings of the 5th European conference on Computer systems*, pages 363–376, 2010.

[134] Xiaohong Zhang and Xiaofeng Chen. Data security sharing and storage based on a consortium blockchain in a vehicular adhoc network. *IEEE Access*, PP: 1–1, 01 2019. doi: 10.1109/ACCESS.2018.2890736.

[135] Muhammad Firdaus and Kyung-Hyune Rhee. On blockchain-enhanced secure data storage and sharing in vehicular edge computing networks. *Applied Sciences*, 11(1):414, 2021.

[136] Gyanendra Prasad Joshi, Eswaran Perumal, K Shankar, Usman Tariq, Tariq Ahmad, and Atef Ibrahim. Toward blockchain-enabled privacy-preserving data transmission in cluster-based vehicular networks. *Electronics*, 9(9): 1358, 2020.

[137] Yuwen Pu, Tao Xiang, Chunqiang Hu, Arwa Alrawais, and Hongyang Yan. An efficient blockchain-based privacy preserving scheme for vehicular social networks. *Information Sciences*, 540:308–324, 2020.

[138] George F Riley and Thomas R Henderson. The ns-3 network simulator. In *Modeling and tools for network simulation*, pages 15–34. Springer, 2010.

[139] https://www.nsnam.org. Network simulator ns-3.

[140] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

[141] Sobhan Moosavi, Mohammad Hossein Samavatian, Arnab Nandi, Srinivasan Parthasarathy, and Rajiv Ramnath. Short and long-term pattern discovery over large-scale geo-spatiotemporal data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2905–2913, 2019.

[142] Steven J Vaughan-Nichols. Achieving wireless broadband with wimax. *IEEE computer*, 37(6):10–13, 2004.

[143] Zhe Yang, Kan Yang, Lei Lei, Kan Zheng, and Victor CM Leung. Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet of Things Journal*, 6(2):1495–1505, 2018.

[144] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[145] Greg Slepak and Anya Petrova. The dcs theorem, 2018.

[146] Sarah Ali Siddiqui, Adnan Mahmood, Quan Z Sheng, Hajime Suzuki, and Wei Ni. A survey of trust management in the internet of vehicles. *Electronics*, 10(18):2223, 2021.

[147] Umar Farooq Minhas, Jie Zhang, Thomas Tran, and Robin Cohen. A multifaceted approach to modeling agent trust for effective communication in

the application of mobile ad hoc vehicular networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(3): 407–420, 2010.

[148] Félix Gómez Mármol and Gregorio Martínez Pérez. Trip, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks. *Journal of network and computer applications*, 35(3):934–941, 2012.

[149] Zhaojun Lu, Wenchao Liu, Qian Wang, Gang Qu, and Zhenglin Liu. A privacy-preserving trust model based on blockchain for vanets. *IEEE Access*, 6:45655–45664, 2018.

[150] M Divya and Nagaveni B Biradar. Iota-next generation block chain. *International Journal Of Engineering And Computer Science*, 7(04):23823–23826, 2018.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Étude et conception d'une gestion décentralisée des données du trafic routier basée sur la Blockchain dans un réseau VANET (réseaux ad hoc de véhicules)

Mots clés: véhicules autonomes, réseaux ad hoc véhicules (VANET), sécurité des données routières, Blockchain, preuve de travail (PoW), tolérance de panne byzantine pratique (PBFT)

Résumé: La prolifération des véhicules autonomes a imposé la nécessité d'une gestion plus sécurisée des données du trafic routier (c'est-à-dire les événements liés aux accidents, l'état de la circulation, le rapport d'attaque, etc.) dans les réseaux Ad hoc pour véhicules (VANET). Les systèmes centralisés traditionnels répondent à ce besoin en exploitant des serveurs distants éloignés des véhicules. Cette solution n'est pas optimale, car les données relatives au trafic routier doivent être distribuées et mises en cache de manière sécurisée à proximité des véhicules. Cela améliore la latence et réduit la surcharge sur la bande passante du réseau de communication.

La technologie Blockchain est apparue comme une solution prometteuse grâce à sa propriété de décentralisation. Certaines questions restent néanmoins sans réponse. Comment concevoir une validation appropriée des données du trafic routier par blockchain, qui semble plus complexe qu'une transaction financière? Quelles sont les performances attendues dans les scénarios VANET?

Cette thèse offre des réponses à ces questions en concevant une gestion des données du trafic routier adaptée aux contraintes imposée par la blockchain. La performance ainsi que la validité des protocoles proposés sont ensuite évaluées à travers diverses simulations de scénarios pris d'un trafic routier réel.

Nous proposons d'abord une adaptation du mécanisme de consensus Preuve de Travail (PoW) dans un réseau VANET, où les infrastructures situées aux bords de routes (RSUs) maintiennent une base de données décentralisée des données du trafic routier. Ensuite, une évaluation rigoureuse des performances en présence de véhicules malveillants est réalisée. Les résultats ont montré que le schéma proposé permet de construire une base de données sécurisée et décentralisée des données du trafic routier au niveau des RSUs.

Ensuite, motivés par nos résultats, nous utilisons PBFT (Practical Byzantine Fault Tolerance), un mécanisme de consensus établi grâce au vote, pour réduire la latence dans le processus de validation dans une blockchain. Les RSUs validatrices de données de trafic sont sélectionnées dynamiquement en fonction de la localisation des événements du trafic. Nous proposons un nouveau schéma de réplication de la blockchain entre les RSUs. Cette réplication choisit un compromis entre les performances en termes de latence et la fréquence de réplication des blocs de la chaine. Les résultats de simulation montrent de meilleures performances, lorsque les RSUs validatrices, sont réduites au minimum.

Dans la dernière partie de la thèse, nous proposons un modèle de confiance pour réduire au minimum le nombre de validatrices sans compromettre la décentralisation et l'équité de la création de blocs. Ce modèle de confiance s'appuie sur la distance géographique et la confiance des RSUs pour former dynamiquement un groupe de validateurs pour chaque bloc de la chaîne. Nous formalisons et évaluons ce modèle de réputation, en considérant divers scénarios avec des RSUs malicieuses. Les résultats démontrent l'efficacité de la proposition pour minimiser le groupe de validateurs tout en isolant les RSUs malicieuses.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Study and Design of Blockchain-based Decentralized Road Traffic Data Management in VANET (Vehicular Ad hoc NETworks)

Keywords: Autonomous Vehicles, Vehicular Ad hoc NETworks (VANET), Road Traffic Data Security, Blockchain, Proof of Work (PoW), Practical Byzantine Fault Tolerance (PBFT)

Abstract: The prominence of autonomous vehicles has imposed the need for more secure road traffic data (i.e., events related to accidents, traffic state, attack report, etc.) management in VANET (Vehicular Ad hoc NETworks). Traditional centralized systems address this need by leveraging remote servers far from the vehicles. That is not an optimal solution as road traffic data must be distributed and securely cached close to cars to enhance performance and reduce bandwidth overhead. Blockchain technology offers a promising solution thanks to its decentralization property. But some questions remain unanswered: how to design blockchain-adapted traffic data validation, which is more complex than an economic transaction? What is the performance in real-world VANET scenarios?

This thesis addresses those questions by designing blockchain-adapted traffic data management. The performance analysis and the validation of the proposed schemes are conducted through various simulations of real scenarios.

We first adapt the PoW (Proof of Work) consensus mechanism to the VANET context whereby the RSUs (Road Side Units) maintain the decentralized database of road traffic data. After that, the proposed scheme is evaluated in the presence of malicious vehicles. The results show that the proposed approach enables a secure and decentralized database of road traffic data at the RSUs level.

Next, motivated by our findings, we adopt PBFT (Practical Byzantine Fault Tolerance), a voting-based consensus mechanism, to reduce the blockchain latency. The traffic data validators are dynamically selected based on traffic event appearance location. Finally, we propose a novel blockchain replication scheme between RSUs. This scheme offers a trade-off between the blockchain latency and replication frequency. Simulation results show better performance when the validators (i.e., RSUs) are minimized.

Finally, we propose a trust model to minimize the validators without compromising the decentralization and fairness of block-creation. This trust model leverages the geographical distance and the RSUs trust to dynamically form a group of validators for each block in the blockchain. We formalize and evaluate this trust model, considering various scenarios with malicious RSUs. Results show the efficiency of the proposed model to minimize the validators group while isolating malicious RSUs.