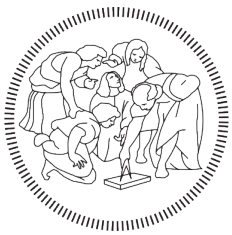


POLITECNICO
MILANO 1863

Scuola di Ingegneria Industriale e dell'Informazione
Dipartimento di Elettronica, Informazione e Bioingegneria
Master of Science in Computer Science and Engineering

Analysis of a Distributed Ledger Framework for Automotive Positioning Applications

Advisor:
Prof. Stefano Zanero
Co-Advisor:
Stefano Longari

Master Graduation Thesis by:
Renato Legler
Matr. 899404

Academic Year 2018–2019

Abstract

Late technological developments allowed processors to be smaller and more powerful, granting devices on which they are embedded advanced computational capabilities. The automotive environment is not an exception, vehicles have become smarter, autonomous and capable of exchange information. Although the great potential of these innovations, traffic management has not yet received significant improvements from these new technologies. In fact, there are not that many frameworks designed to assist traffic, and the existing ones are highly centralized and company-owned. Centralization, in particular, if a company is the centralizing entity, brings different threats: on one side the company can be subject to cyberattacks, and on the other, the company itself may not work in the attention of the end users, regulating policies against their interest. The purpose of this thesis is to provide a detailed feasibility study of a new framework, designed to manage information exchange between vehicles through the use of a structure based on a distributed ledger. The framework adopts a consensus algorithm to allow users to evaluate the truthfulness of the information shared within the system, which grants them different values of reputation exploiting statistical inference to assess the reliability of the information. The information, in the analyzed scenario, comprises the position of the sender alongside the ID of a second vehicle. The objective is that of tracking the location of the participants by correlating the information sent by all while keeping a reputation score to detect malicious agents. The implementation of a simulator allowed to conduct experiments that proved both the feasibility of the system in real dimensioned scenarios and its resistance against possible cyberattacks. The research provides ground for future studies since the fast paced development in semi-autonomous vehicles from all manufacturers will shortly lead to the necessity of developing new, company-independent technologies to enable communication. At the same time, the increasing amount of analyzable data in the short future will surely help in increasing the feasibility details of the proposed framework and similar ones.

Sommario

Le novità in campo tecnologico degli ultimi anni hanno portato i componenti di dispositivi ad essere non solo sempre più performanti, ma anche più piccoli. Questo ha permesso loro di essere installati in apparecchiature che diventano ora capaci sia di interconnettersi, che di funzionare in modo assolutamente autonomo.
Molti di questi dispositivi, che fino a pochi anni fa era impensabile potessero avere dei computer a bordo, sono ora in grado di comunicare ed elaborare richieste.

Diversi campi sono stati influenzati dalle conseguenze di queste migliorie tecnologiche, ed il settore automobilistico non è rimasto escluso. I veicoli più all'avanguardia, infatti, sono oggi equipaggiati con decine di processori che assistono il guidatore in numerosi compiti, dalla guida autonoma all'identificazione del percorso migliore. Nonostante le numerose innovazioni, però, i servizi e le applicazioni volte alla regolamentazione del traffico, visto come l'insieme dei veicoli circolanti, non hanno ricevuto lo stesso interesse ed utilizzano ancora tecnologie ormai superate.

I pochi sistemi esistenti che si propongono di fornire questo servizio sono fortemente centralizzati e quindi l'intera responsabilità viene affidata ad un solo ente che può essere soggetto ad attacchi informatici. Le minacce esterne, tuttavia, non sono l'unico problema, infatti, se a capo di un servizio vi è una sola azienda, questa sarà in grado di rimodulare le proprie politiche a piacimento, o, addirittura, impedire a determinati utenti di usufruire del servizio.

È qui che nasce l'idea del framework da noi proposto. Lo scopo della tesi, infatti, è quello di presentare un sistema decentralizzato, distribuito e aperto a tutti. Grazie ad una struttura basata su un registro distribuito, i partecipanti possono scambiarsi informazioni riguardo la propria posizione geografica e quella dei veicoli che hanno attorno.
La percezione degli utenti nelle vicinanze è ottenuta grazie ad una comunicazione a breve raggio, DSRC, mentre per accedere alla blockchain è possibile utilizzare quella cellulare, 4G o 5G.

In un ambiente dove ogni utente può condividere informazioni avendo la medesima autorità, secondo i principi della blockchain, eleggere un validatore e valutare correttamente quando un'informazione trasmessa è vera o falsa, può non essere così semplice. Non essendoci un ente centrale incontestabile, spetta ai partecipanti stessi assicurarsi un corretto funzionamento della rete; viene quindi fornito loro un algoritmo per determinare la veridicità delle informazioni condivise.

Grazie ad un'inferenza statistica sulle posizioni dichiarate dagli utenti, esso è in grado di valutare se la località dichiarata da un partecipante possa essere o meno attendibile. Gli utenti, infatti, hanno un valore inizialmente uguale che rappresenta la loro reputazione, questo numero varia in base all'autenticità delle informazioni dichiarate. Se esse saranno concordanti con quelle di altri utenti, il punteggio crescerà, in caso contrario un utente che ha mentito verrà penalizzato e le sue future dichiarazioni saranno ponderate con più diffidenza. Basandosi sulle reputazioni equilibrate in questo modo, vengono scelti anche i validatori dei blocchi, che ogni 60 secondi raccolgono e verificano tutte le informazioni scambiate.

A causa del difficile adattamento dei software esistenti alla struttura da noi proposta, è stato implementato un simulatore che ricrei diversi scenari regolati con questo approccio. Grazie ad esso è stato possibile eseguire delle sperimentazioni per studiare meglio il comportamento del framework e le reazioni inaspettate.
I risultati ottenuti provano la fattibilità del progetto. Le soglie minime e massime per diversi fattori, come la dimensione della mappa, la mole di messaggi scambiati, il numero o la densità dei partecipanti, sono compatibili con situazioni reali. Inoltre, sono stati condotti dei test per valutare la resistenza del sistema di fronte sia a situazioni limite che a possibili attacchi

informatici, volti a causare un malfunzionamento del servizio tramite false informazioni.

Gli studi portati avanti in questa tesi possono fornire le fondamenta per ricerche future nell'ambito considerato, tenendo presente anche che la diffusione di veicoli intelligenti aumenterà il numero di dati statistici a disposizione, utili a rifinire i parametri proposti. Un'effettiva implementazione di questo progetto potrà portare non solo un cambiamento nel modo di percepire il traffico, ma anche una diversa assistenza ai mezzi di emergenza, a veicoli con guida autonoma e una maggiore sicurezza di tutti i mezzi circolanti su strada.

Contents

List of Figures

List of Tables

List of Listings

Introduction

Computer Science continues to bring radical changes and countless new technologies over time. Processors can now be components of objects that a while ago could not be conceived to be equipped with computer-like elements. Progresses like these are possible due to embedded systems that are getting smaller, making even uncommon items capable of great computational power. Technologies are not only advanced but they are also able to communicate amongst them.
The study carried out in this thesis, indeed, has been run in a period that features technologies that are more and more interconnected, but, at the same time, are developing an impressive autonomy.

These technological enhancements have also affected the automotive scope, in which vehicles are adopting embedded systems granting them high-performance capabilities. Just think of a Sat Nav that seemed to be a revolution a few years ago, and now car are equipped with tens of CPUs[1] components [1], that make them able to elaborate requests and exchange messages.
We can just assume what will be the impact of such a swift technological improvement if applied to the most used vehicle in the world [2]. In no time, as was for many other technologies, our perception of car transport and vehicles, in general, will be completely revolutionized.

[1] CPU: central processing unit

In this context, though, systems that assist traffic management have always been using the same technologies, they often consider only the individual vehicle they are installed into, and not the totality of nearby road participants. Moreover, the few existing systems concerning this scenario are strongly centralized and controlled by big companies such as Google [3], [4].
Centralization itself may not be an issue, but an approach of this sort entrusts all the responsibility and trust to a single entity, subjected to possible attacks and own policy regulations that could cause difficulties to users or even restrict the access to certain categories at the discretion of the company that offers the service.
Furthermore, these applications rely entirely on cellular network, most are not even connected to the onboard unit of the vehicle, and are used either for data collection or information acquisition, leaving the structure exposed to information falsification and invalidation of a proper service.

The framework we propose, on the other hand, is designed to be decentralized, distributed and open to everyone. Participants, indeed, can collect information thanks to a short-range communication, harder to forge, and send it on a network based on the concept of distributed ledger. Here, in an environment where every node has the same authority, embodying the radical principle of blockchain, validators are elected to regulate the system; it is, therefore, up to participants themselves to ensure the correct functioning of the system.
We adopt a new algorithm that grants users the opportunity of evaluating when a piece of information shared through the service is reasonably true or rather fake, basing on a statistical inference on previously collected data and the assessed reliability of their sender.
An implemented simulator allowed us not only to better calculate thresholds and dimensions but also the system response to possible attacks, whether they are organized or not.

The results obtained through testing prove the feasibility of the framework: the limit thresholds of users distribution are compatible with real situations that feature overcrowded cities as well as less populated area.

Also, tests to analyze attack resistance suggest a proper defense and reasonable robustness.
The research provides ground for future studies; automotive technologies will gradually improve and spread, meaning that data will be easier to collect and analyze. An actual implementation of this framework will lead not only to renovated traffic management but could also assist services that deal with it, such as emergency administration, autonomous vehicles, taxi services and safety in general.

The thesis is structured as follows:

- ◇ In Chapter 1 the contextualization of the study and the scope of the research are presented. The state of the art is described, and so are all the elements and components that will be subject to further analysis. Goals of the study are also defined, together with challenges encountered.

- ◇ In Chapter 2, it is presented a brief overview of what has been done and also a detailed step by step analysis of the conducted studies to reach conclusions. Data collection, elaboration, the consensus algorithm, how support tools were implemented, dimensioning, initial condition and eventually a simple use case will be also discussed in this section.

- ◇ Chapter 3 is dedicated to the description of the implemented simulator, used to verify the direct application of the algorithm. Here are presented the class and main methods, then the control flow is analyzed in details.

- ◇ Chapter 4 contains all the experiments stated in detail. The procedure to set up the dataset is also described, along with the motivation of all design decisions. Tests are divided into two types, one dedicated to the feasibility, the other to study the consequences of possible attacks.

- ◇ Chapter 5 is dedicated to the discussion of challenges faced during the research and which limitations we had to confront with, mostly due to the state of the art.
- ◇ Eventually, Chapter 6 is the closing one. Here are recalled obtained results and what we dealt with; finally, some speculation on future studies and prospects are expressed.

Chapter 1

Motivation

1.1 Problem Statement

The latest progress in communication and automotive development led to the implementation of useful technologies and smart ways to deal with them. Cars are equipped with powerful and often autonomous components that allow them to reach a considerable awareness of their surroundings. Although vehicles are achieving more and more performing devices, the same cannot be said regarding traffic management tools. Not only there is a lack of frameworks that provide such services, but the existing ones are private, centralized and managed by a singular company.
This means that in any moment, the operating corporation could change its regulations, deny access to the service to certain individuals at will, raise any price, shut down servers or the entire business, be the target of cybercrime and other threat that would have severe consequences on user experience. Not to mention expensive commission fees, privacy violations, frauds or censorship that commonly mine stability of centralized online service providers.

The research carried on in this thesis aims to explore in greater depth, the feasibility of a proposed framework that connects the concept of distributed ledger to the automotive industry; analyzing the conditions

of a successful functioning, the requirements and the trade-off between efficiency, security, and scalability.

The introduction of a structure based on a distributed ledger will provide a series of advantages, interesting not only from a computer science viewpoint but also in the sense of reallocation of responsibility and decisional power, carried by the distribution property of this configuration. Decentralization, indeed, is a revolutionary characteristic when dealing with services provided publicly. The system is open and accessible to anyone, technically always. There is not a central corporation that manages traffic or personal data, and all participants are potentially equal. Furthermore, the emergence of smart vehicles is probably going to advance even more in the next years; studies dealing with this topic will be soon really useful.

1.2 Technological Background

The study is the analysis of a proposed framework to share messages among intelligent vehicles equipped with an appropriate computer system. Messages contain information regarding the reciprocal position of the entities involved and can be publicly consulted in order to gather useful information about the current traffic situation. The structure of message exchange is based on the concept of blockchain.
The significant expansion in the usage of technologies relying on a distributed ledger has encouraged the adoption of this structure in the framework proposed. Adopting the blockchain opens up the possibility to create, manage and access a public ledger shared among all participants.

1.2.1 Blockchain Structure

Usually, data exchange and communication have always been handled by a trusted intermediary, that manages and tries to guarantee security. However, several concerns arose regarding the reliability of the intermediary, hence P2P[1] communication became an appealing alternative. This is

[1]P2P: Peer-to-peer

when the idea of blockchain was born [5].
It is important to understand that Blockchain is not an actual technology, but rather, a data structure to be adopted in a system; and different typologies can be implemented changing the characteristics of its components. The main units are:

- **Nodes**, the agents of the chain that contribute to generate transactions.
- **Transactions**, contain information about participants' location.
- **Blocks**, collect transactions every certain period of time and are validated by a Node.
- **Ledger**, it is the public result of all transactions, it is shared among every participant on the chain.
- **Asymmetric Key Cryptography**, it is fundamental to ensure the authenticity and integrity of transactions.

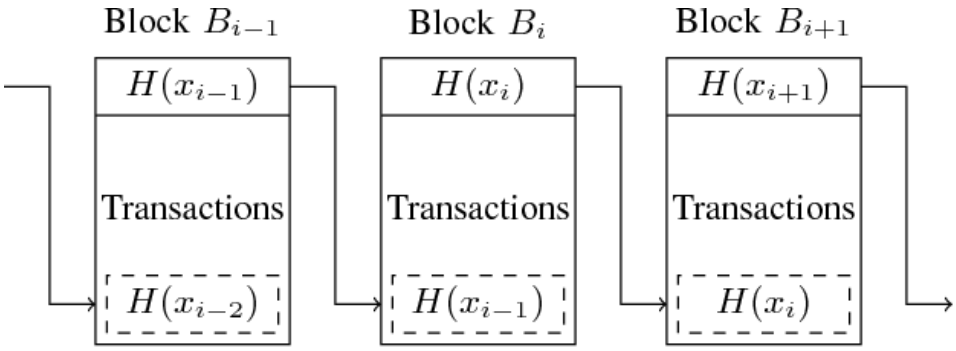


Figure 1.1: Blocks are made of transactions and metadata referring, also, to the hash of the previous block [6].

First of all, the most relevant advantage brought by blockchain is decentralization. As said, this structure can eliminate the need for any intermediary, there is not a central entity who regulates or takes decisions: everyone has the same authority in the network. Any validated transaction will be stored in the blocks of the chain, anyone can access them, anytime, making the network absolutely transparent and trackable. The distribution of the ledger makes it somehow more attackable since every node can be a victim, but at the same time it is also much more robust because the information is shared among all the participants and its corruption requires the takeover of the greatest part of the network. Once a block has been validated, the transactions in it contained cannot be modified without the whole blockchain noticing its editing. Eventually, it is also possible to implement automatic reactions into the chain, that activate upon the occurring of a certain event or condition, allowing the network to be reactive and independent [7].

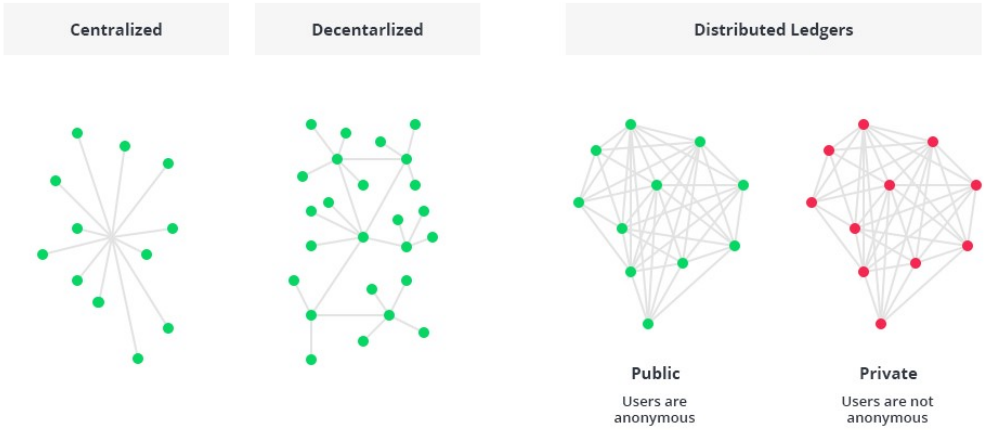


Figure 1.2: A visual representation of different ledgers types.

The smallest parts that contribute to the chain composition are what blocks are made of: transactions. The exchange consists of files containing transfer information, generated by a source node and broadcast to the entire network. They constitute the current state of the blockchain itself, once they are congregated in blocks. When dealing with cryptocurrency, transactions represent the transfer of tokens from one node to another, but every participant is aware of every transaction in order to maintain

a consistent log with the whole history of exchanges. Each transaction contains the digital signature and the public key of the two parts involved. Once being broadcast, every transaction issued after the last block validation is collected into a new block ready to be examined by chain participants. The elected validator, chosen according to the blockchain policy, checks those block transactions and, if the information stored are considered being genuine, they are authorized, and the validator adds its ID and the calculated hash. Now that the block is validated, it can be added to the public chain and it is available to every participant. The state of the entire blockchain changes after the validation of each transaction [8].

Depending on the access mode, blockchain can be classified as public, private or consortium. Public blockchains do not impose any restriction to access, potentially anyone can join and issue at any time. While a completely decentralize chain may seem more exposed, the high number of participants together with the cryptographic validation and a robust consensus algorithm can mitigate this exposure. A system that makes use of a private blockchain can control accesses and mining; it is more centralized and usually the possibility of consulting the ledger is granted to every participant, whilst writes are restricted. Eventually, a consortium blockchain is a mixed protocol where no single organization is responsible for validation and access regulation, but a set of nodes is, making the system partially centralized and permissioned, but openly accessible.

Property	Public blockchain	Consortium blockchain	Private blockchain
Consensus determination	All miners	Selected set of nodes	One organization
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus process	Permissionless	Permissioned	Permissioned

Figure 1.3: Characteristics and properties of the different policies that can be adopted in a Blockchain [9].

1.2.2 Related Implementations

The issue of privacy became more and more important, especially in the last years, the impact of data has become known and so its risks. Blockchain could solve by design many uncertainties, taking advantage of the interconnections between smart vehicles and roadside infrastructure, their proximity and encryption to reduce odds of privacy or security breaches [10], [11].

The concept of blockchain is usually associated with the idea of cryptocurrencies, in particular, with the popular Bitcoin, expressed for the first time in its whitepaper [5]. The idea of timestamping transactions by hashing them into an ongoing chain of hash-based calculation, forming a record that cannot be changed without redoing the validation was actually thought by Satoshi Nakamoto in the first place, to power the cryptocurrency.

Afterward, however, several different uses have spread for this technology, as [12] states. From the voting system used for the first time in West Virginia to Real Estates and land properties to track owners history, used for identity solutions for refugees, insurances or certificates. Also, in Italy, it has started to be implemented in finance, insurance, payments, health care and tracking food origin monitoring its supply chain, let alone industry 4.0 and IoT[2].
Blockchain has also been utilized in automotive scope [13]. As a solution to threats like hijacking, tracking or exposed vulnerabilities of smart vehicles. Software updates or other emerging services such as dynamic vehicle insurance fees are often performed through vulnerable wireless communication; blockchain can enhance security in this scope.

An interesting study [14] showed how blockchain can be used to verify that locations claimed by its users are their actual one. This is particularly helpful in systems that are based on geographic information and points out how centralized verification approaches proposed in the past are not so applicable since may represent a high risk in privacy for users. That

[2]IoT: Internet of Things

framework makes use of schemes to avoid fake location forgery and, thanks to Bluetooth enabled users they can generate mutual proof of their location.

Eventually, blockchain has also been used in supply chains to precisely know the origin of components. Also, Reply has launched "That's mine", a project aimed to identify the current owner of a certain asset: the decision was made to develop a service dedicated to one of the specific areas in which the certification of ownership has great value associated with it, opting for the automotive market. Vehicles were a pertinent target due to their widespread and need to verify ownership, their management is centralized by now [15]. Introducing Blockchain in this field could lead to an immediate change of the owner, public access to ownership verification, enhanced security and costs cut.

1.2.3 Consensus Algorithms

The fundamental distinction amongst various kinds of blockchains can be found in the manner in which consensus, concerning an information, is reached. Determining which information should be shared among everyone is vital. Numerous strategies can be adopted to reach the agreement.

The most famous is Proof-of-Work, which is the one used by Bitcoin blockchain. The involved nodes who want to validate the next block and gather some extra token compete to resolve complex mathematical riddles. It is very hard to reach a solution, but on the contrary, it is easy to check its truthfulness. The difficulty can be changed to adapt to the current needs of the chain. This algorithm is slow and very performance requiring, but it is stable, trusted and widely applied [16]. A variation of this method is the Delayed Proof-of-Work, a hybrid protocol that allows a chain to take advantage of the hashing power of another blockchain, making it more energy-efficient. However, there is growing evidence about the increasing demand to solve blocks and the bandwidth overhead are getting too high to be suitable for smaller or embedded systems. Internet of Things devices arisen this issue first, due to their limited computational capacity. The network presents also scalability problems and high delays to reach consensus [17].

Raft algorithm, proposed in [18], reaches consensus through a leader election; there are leaders, followers, and candidates. A Leader keeps broadcasting its presence every 150-300 ms, if a follower does not receive this "heartbeat", it applies itself for the election. This procedure is secure and implementable in many languages but is usually adopted by private or permissioned networks.

There is also another, different, kind of chain, where blocks are not added one by one sequentially but rather simultaneously. Several algorithms apply this policy and vary upon the decision of how to validate previous bocks or transactions, the order, and final validation achievement. Although requiring a more complex system, it can lead to a great scalability due to nonlinear structure, high transactional throughput, energy-saving, and almost instant validation. Some structures that followed this lead are the IOTA's Tangle, Hashgraph Blockchain, Holochain, Block-Lattice, ByteBall and SPECTRE[3] [19], [20].

To conclude the list, the Proof-of-Stake is a pseudo-random election process, used to draw the validator of the next block. There are many variations of this design that leverage on different properties to drive the odds. Ethereum is a well-known cryptocurrency that recently switched from Proof-of-Work to this algorithm [21]. Validators are still awarded the fees of the transactions contained into the mined block but, in this case, participants do not rely on computer power but on a stake bet to win a sort of lottery. Participants, indeed, put at stake a number of their tokens to improve their odds to be drawn as the next validator, with all the benefits of this win. The amount staked is locked away until a validator is extracted. This solution provides efficiency and no computational waste but may grant too much power to the richer participants.

Many new consensus algorithms were born as a variation of the above-mentioned structure. Granting more importance to the coins that were cycling for a longer time, burning the tokens staked or even staking different valuables. In this last category fall in the main algorithms analyzed.

[3]Serialization of Proof-of-Work Events: Confirming Transactions via Recursive Elections

- ◇ **Proof-of-Importance** is based on Proof-of-Stake but, in addition to the stake, fame, balance and past transactions of the node are considered [22].

- ◇ **Proof-of-Reputation** is a protocol designed to host candidates for validation that will have to face heavy financial or reputational losses in case of transaction falsification, this is their stake [23].

- ◇ In **Proof-of-Authority**, the validation process is entrusted to private accounts committed to keeping their computers not compromised; the eligibility to this charge is difficult to obtain but provide economic benefits [24].

- ◇ The implementation of **Proof-of-Believability** includes the division of participants into two groups: a trusted League and a normal one. When normal users that make transactions and validate blocks, accrue points of reputation so that to overcome a certain threshold, they access to the trusted League; here, they benefit from a faster validation of their transactions. IOST is a cryptocurrency that implements this method, Servi is a supervisor software that calculates the distribution of reputation [25].

There are a lot of advantages introduced by this structure and security is an aspect which is being affected. Cybersecurity, indeed, is a topic achieving more and more attention among fields not solely regarding computer science. Automotive security is currently undertaking in-depth studies to reach specific requirements and is discussed by many experts in this area [26], [27].
A previous study [28] has been proposed by Politecnico di Milano, analyzing the suitability of the use of blockchain in vehicular settings, highlighting how the implementation of this recent technology could be used to reduce threats that are jeopardizing V2X[4] environment.

[4]V2X: Vehicle to everything communication

1.2.4 Blockchain Disadvanteges

It should be taken into account that blockchain is not a flawless innovation: shortcomings and trade-off must be considered. Depending on the number of participants and the consensus algorithm adopted, the blockchain can encounter difficulties in scalability that may even lead to its collapse.
If the validation agreement is not reached in a short time, the system may not achieve its purpose; Bitcoin, for example, has a block time of about 8 minutes [29] that is extremely slow compared with the few seconds of a credit card transaction. Proof-of-Work is no longer the best option when it comes to high transaction throughput, and even worse when energy consumption is considered. The computational requirement, indeed, is immense: 330k times less efficient than a common Visa transaction [30], that is not only a problem of cost but also environmental. Blockchain dimension must also be considered because to maintain the chain authentic, the whole transaction history must be recorded and kept stored somewhere by every participant. Bitcoin blockchain size is nowadays about 236 GB [31], meaning that every new node must download it all before committing a transaction, creating also storing difficulties.

1.2.5 Automotive scope

Vehicles, that will be the intermediary between the framework and participants using it, are evolving into autonomous mobile-connected platforms. Cars will be more and more equipped with advanced information and communication technologies that are already being introduced into modern passenger vehicles. They are built with devices granting them intelligent capabilities.
This has resulted in the introduction of perception sensors utilizable in traffic situations and technologies that are advancing from simple and only informative, toward the control of the entire vehicle. The introduction of wireless links among vehicles should enable the sharing of information and thus enlarge the situational awareness of drivers, as the perceived area is

enlarged.
Vehicles are inserted into a cooperative network because onboard sensors alone cannot observe and thus grant functions that allow the deployment of vehicles with autonomous capabilities, on the other hand, information achieved from the network can be wider and gathered by multiple viewpoints. The availability of intelligent sensors, advanced digital maps, and wireless communications technologies together with the availability of electric vehicles should allow for deployment on public streets without any environmental modification. Likely, there will first be self-driving cars followed by environmental changes to facilitate their deployment [32].

Vehicles can employ a variety of wireless technologies to communicate with other devices, being them infrastructures or vehicles in turn. One of the most common in the vehicular application is Dedicated Short-Range Communication, which is presented in detail in [33]. The primary motivation for DSRC deployment is collision prevention since it allows vehicles and infrastructures to exchange data frequently and promptly. But this protocol can also be exploited to share other types of messages, like traffic information, weather or existence of hazards. If a vehicle determines that a potential collision is occurring, the On-Board Unit can take action to warn the driver or assisting the vehicle control. While the communication between DSRC devices must follow carefully designed interoperability standards, how messages are digested and used is up to the automobile manufacturer. The term Short Range in DSRC is meant to convey that the communication takes place over hundreds of meters, compared to other communication networks.

1.3 Goals and Challenges

The scope of the thesis is, therefore, to take advantage of all these emerging technologies and connect them into a single, useful, system. Traffic management services or applications exist nowadays, but they are centralized, meaning that only one entity is administrating them. This leads to a centralization of the power into a single unit that can manage

independently the functioning of the system. The access to the system could suddenly become private due to the policy of the company that runs it, charges may be applied on a platform that has been free so far. Moreover, no matter how trustable an intermediary is, it always represents a single point of vulnerability. Delegating the security responsibility to a single system can be convenient sometimes, but the list of attacks a company is subjected to, causing data breaches or economic damages, is endless.

A framework like the proposed one has not been studied in depth yet. A self-managed algorithm, capable of evaluating the reliability of users without a unique coordinator that decides who should be rewarded or penalized, whose authority cannot be questioned, is what the state of the art has not faced yet, not in automotive scope at least. The goal is the introduction of a process that allows every user to participate in a network, having the possibility of exchanging information and develop credibility on which other users will rely on to believe each other or not.

The outcome of this study could lay the foundation to further studies in the direction of decentralized traffic management. The results and data provided could be used to draw up a guideline to implement a system that does not currently exist and refine it.

Right for these reasons, the progress of the study encountered some difficulties such as the lack of previous results, data, samples, and comparisons that made sourcing more challenging. The construction of a dataset with information regarding accesses and city dimensioning was not trivial. Also, no existing software turned out to be useful, whether because of their impossibility to be adapted to our needs or the complexity of the code that would have required a complete redesign of it.

Chapter 2

Approach

2.1 Approach Overview

The research is a feasibility study of a framework aimed to manage information exchange among vehicles, sharing massages on a blockchain. The study stems from a previous paper [28], that proposed a primitive suggestion of this idea. The first step has been to ensure the novelty of the framework as it was thought. Then, each "component" of the system was analyzed alone: vehicles, messages, blockchain, consensus algorithm. Previous studies and data were consulted in order to find possible starting points to design the framework. Two types of communication were taken into consideration to interact with the network, a short-ranged one aimed to gather proximity information about near vehicles, and a long-ranged one, to send packed information into the chain.

Several typologies of blockchains and consensus algorithms were analyzed [5], [19]–[21], [34] in order to find one that would satisfy all the requirements and could provide the best trade-off in security and average computational effort, but above all scalability, due to the numerous record of potential users. An existing company, Internet Of Services foundation, proposed an innovative blockchain paradigm achieving high throughput, security, and scalability. The structure at the base of their cryptocurrency, the IOSToken [25], and the main components that made the idea so valu-

able are the solid consensus algorithm based on Proof-of-Believability, the possibility of dividing the network in separated but synchronized shards, and the concept of Micro State Block.
Their consensus algorithm was studied and, with some variations, adapted to the needs of the study. The algorithm is not only applied to pick the validators of subsequent blocks but also to assess the truthfulness of the information shared. Users, indeed, have a personal reputation score that can vary in accordance with their transactions: if a user shares an information that has been deemed to be truthful, they are rewarded with an increment of their reputation, otherwise if the information is discordant with what has been stated by other participants, making that statement most likely incorrect, they are penalized.
Intuitively, a user with a high value of reputation is trusted more, when sharing information, compared to a lower one. This is the basis of statistical inference calculated to distinguish genuine and malicious users.

As regarding the proper mechanism of functioning, participants probe their vicinity to intercept other nodes, be they vehicles or roadside stations, and generate a transaction with their location and the distance measured from the target. The transaction is then broadcast to all the nodes in the same shard.
Every minute a validator is extracted, according, also, to their reputation level, then they proceed to pack all received transactions during the last 60 seconds into a block and validate it, trying to deduct who lied and who told the truth. Two supports are elected too, to check the validity of the new block, in the event of a malicious validator. Every quarter of an hour, each shard elects a representative to perform a similar election and synchronize blocks on the whole network.
Possible scenarios were analyzed and, through the implementation of a simulator, a hypothesis of the operative functioning of the framework was tested. The initial condition to assist the system in reaching autonomy is further discussed.
Eventually, several attacking scenarios were tested through which limitations and security thresholds are proposed.

2.2 Approach Details

2.2.1 Background Analysis

Firstly, blockchains, whether existing or still theoretical, were classified according to their main parameters and eventually elected as possible choices.
One of the most valuable characteristics in the scope of the thesis is the throughput: the number of transactions that a blockchain can issue in a second; so is the blocktime: time span from the validation of a block to another. The blocksize is also a good assessment criterion: the maximum number of transactions that can be wrapped into a single block; finally, the active number of nodes that the system can manage.
The strict requirements of the framework allow to reject in the first place many options: private blockchains could not be the solution, nor those aimed to manage a small number of users.
The traffic situation must be fresh and reactive: a high throughput of transactions is necessary, this leads to consider only fast and even asynchronous blockchains.
Although computers embedded on smart vehicles are getting more and more powerful, the condition of consensus cannot be a Proof-of-Work due to its extreme complexity which happens to be too time and energy-consuming.

An initial idea, took into consideration the concept of the Tangle, an asynchronous directed acyclic graph, just made of transactions and not blocks. This typology removed the necessity of miners since it does not require a proper block validation but only the validation of a couple of transactions before committing one, self-fueling [35].
The Tangle is extremely fast thanks to the asynchronization of transactions that allows to commit several at the same time. An analysis of its defense to attacks corroborated its implementation in the framework.
But, unfortunately, a concrete example by IOTA, the company that pushed its use, demonstrated how the application of the Tangle required a sort of unquestionable coordinator not only in the deployment phase, as stated at the beginning by developers but was always required to ensure the stability

against double-spending attacks and main chain hijacking. The absolute need of a central entity to manage a decentralized framework was not acceptable and forced the rejection of the idea.

This is when Proof-of-Believability was taken into account. Relying on a Proof-of-Stake approach solves the problem of high computation, allowing every agent to join the blockchain, regardless of their vehicle asset's power. A group of researchers from Harvard, Princeton, University of California at al. proposed an algorithm based on the believability of users and Internet Of Services Foundation invested to distribute a cryptocurrency based on it, the IOSToken.
Other than the use of a stake to validate blocks, many properties of this protocol turn out to be valuable: an extreme high number of transactions per second, providing the basis for a scalable and secure solution; also, the possibility to divide the network into pseudo independent sub-divisions, allowing participants to communicate almost instantly.
To speed up the communication even more, it is applied the concept of Micro State Block, which lets participants with a limited storing capacity to save only the headers of old obsolete blocks, streamlining both the constant update of the chain and the memory required to store the entire state of it. The time window to consider "old" a block can be regulated, since an old transaction already validated, that updated the reputation of its sender, is no longer useful.
IOST counts currently over 344k accounts, 140 million transactions, 409 nodes validators, and 555 smart contracts[1].

2.2.2 Blockchain Design

Every user has a pair of private and public keys, with which they can digitally sign emitted transactions, exactly like happens in cryptocurrency blockchains. This can ensure authenticity of messages, but not necessarily their truthfulness.
Key distribution might be a controversial topic since delegating the responsibility of whether allowing people to join or not the network, to a central

[1]Data acquired on November 26^{th}, 2019

authority, could lead to a permissioned blockchain. In order to enhance security, a compromise between complete autonomy and regulations is considered to restrict the chance of an attacker to obtain unlimited fake devices working as a vehicle: delegating an authority the responsibility for issuing public and private key, hence granting the right to send transactions into the blockchain.
Two entities are considered in this scenario.
First option: empowering *car dealers* to distribute keys. This solution is more open and freer, each license plate has a key linked to and a user has different profiles for each vehicle they own. In this scenario, the key is univocally bound to the vehicle.
A second possibility is represented by empowering *local DMVs*[2] to distribute keys. This idea is more restrictive, key distribution is up to a state authority that also distribute documents as driving licenses and thus deemed more secure. Adopting this protocol, the id and key are bound to the license i.e. the person, making users unique in all their vehicles. In this scenario, it may be more difficult to figure out who is driving, but it should be taken into consideration that, in most cases, each vehicle is registered to a single person.

Transactions are the constituent part of the blockchain, they are shared among nodes and collected into blocks, to be immutably stored.
Their entries are:

⇒ Sender Id

⇒ Target Id

⇒ Current location of the Sender

⇒ Measured distance between the two nodes

⇒ Timestamp of the transaction and a Digital Signature

Once the transaction is assembled, it is serialized in order to be standardly read as it is shared among participants. Considering the average transactions per block [36], the potential scalability, and a trade-off between reactivity and feasibility, a fixed blocktime has been designed to be

[2]DMV: Department of Motor Vehicles, the Motorization

60 seconds.
During this span of time, more precisely 50 seconds to allow late messages to reach everyone, participants generate, share and collect transactions. When time is up, every node calculates three random numbers basing on the unpredictable property of transactions: timestamps and IDs, using a randomization process as hashing repeatedly incoming information. If no node lies, nor transactions are lost, every participant will have calculated the same number, otherwise, numbers are broadcasted, and the majority is selected. The reduction function from last hash to an integer is designed to return a value in a certain range, depending on the number of active users. These have a dedicated opportunity range to be picked as the next validator that depends on their current reputation value. Explicitly, the total sum of active users' reputation is calculated, then it is dedicated them a specific range for every node, depending on the percentage of a node's reputation compared to the total. The next validator is the owner of the dedicated range in which the drawn number fell, while the other two numbers decide the support nodes for the validation. An example is provided in Section 2.2.6 to better understand this process.
It is remarkable how the election result is not deterministic, but at the same time benefits nodes that behave properly; the random nature of this election reinforces the system against organized attacks, as well as a possible centralization in favor of few very trustable participants. At this stage the designed validator checks every transaction received during the past blocktime, it is verified the pair id-signature of the sender, and as standard practice for blockchains, the hash of the block salted with the previous block's gets added.

2.2.3 Reputation and Validation

Due to the design of the network, every participant can share transactions. As has been said, this is a point in favor of the blockchain, but it also represents a huge opportunity for malicious entities to commit false information for their own benefit.
In this context, establishing permissions or limitations is not the solution,

it is therefore necessary to apply a robust procedure of verification. The proposed solution provides for an algorithm that tries to statistically deduce which information is correct and which one is not.
The idea is to rely on the number of users that state a certain location of a vehicle to infer the truthfulness of a transaction having that vehicle as target. All those statements are weighted in accordance with the level of reputation of the node emitting it.
A practical example could be the occurrence of a transaction that declares the position of a vehicle V in a certain location. Several participants, though, with a high reputation value, claimed a few moments ago that V's location is very far from those coordinates, the sender of the first transition is probably incorrect.

The process of verification examines all transactions of the block, collecting, for each, all the other transactions that have the same target of the examined one, in the same block but also in the 20 previous blocks[3], weighted to give more relevance to the latest ones. If the number of transactions involving that target is sufficient, they are divided into two groups, one containing transactions concordant with the analyzed one and the other composed of transactions stating locations that are incompatible. This distinction is made through the following if condition:

```
if
   trans.getLocation().distanceToPos(tempTrans.getLocation())
   ≤
   CONSTANTS.SENSIBILITYDIST +
   trans.getDistance() +
   tempTrans.getDistance()) +
   2 * timePassed * CONSTANTS.MAXDISTANCECROSSED
then
   it is concordant, otherwise not
```

[3]This number is a parameter, and can still be changed, as it will be stated further.

This condition checks if the distance between the current location of the sender of the analyzed transaction and that of the sender of one of those targetting the same vehicle is minor than the maximum distance it can take. Indeed, their distance cannot be more than the sum of the declared one toward the target, plus the diametrically opposed possible distance traveled, plus a small value of possible measurement error.
Afterward, the sum of all the reputations in the two groups is calculated and according to which one is greater, the analyzed transaction will be considered true or not. If the difference of the reputations is near to zero, then the truthfulness is not clear and no nodes reputation is updated.

The algorithm regulates reputations in accordance with the following equation 2.1:

$$\frac{(\Sigma_{i \epsilon I_c} R_i - \Sigma_{j \epsilon I_d} R_j)}{\|I_c\| + \|I_d\|} = x \tag{2.1}$$

Considering:

- I_c the group of agents that have shared a *concordant* information about a certain vehicle.
- I_d the group of agents that have shared a *discordant* information about a certain vehicle.
- R_i the weighed reputation of an agent i at the moment of transaction.
- ε a minimum threshold value to consider an information biased. (the `UPTHRESHOLD`)

Depending on the result of the equation:
if $x > \varepsilon$ reward $i \epsilon I_c$ and penalize $j \epsilon I_d$
if $x <$ - ε penalize $i \epsilon I_c$
if - $\varepsilon \leqslant x \leqslant \varepsilon$ mixed information, stand down

Once all block transactions are verified and nodes have been rewarded or penalized, the validator broadcasts the block to the whole shard. At the same time, the two support nodes, make the same verifications, and, with a certain sensibility threshold, to compensate possible transactions loss, check if the block is correct and reputation changes have been stated properly. In case of correct validation, they broadcast an *ok* message, otherwise a *reject* one. When participants receive the block, they wait for the *ok* from at least one support, then proceed to update their version of the blockchain.

With respect to IOST, some changes are applied: the concept of Servi, a support software to calculate reputations, is replaced by validators themselves. Also, the completely random nature of shards allocation is revisited. Indeed, the protocol includes the division of the network into shards, that are a sort of sub-networks of the blockchain itself.
In this way, each node has a constant access only to a part of the entire chain, this does not mean to lose decentralization, since the chain is still updated at regular time, but allows transactions to be committed in parallel on different shards and speed up the throughput.
Shards are divided according to places geographically near, in order to let nodes in the same shard to access information almost instantly. Decreasing the dimension of the shard and making them less random, there is a risk of facilitating organized attacks. Therefore, we opt to join, under the same shard, different areas picked randomly, so to keep dynamically stable the number of active nodes into the shard.
In this way, the robustness of the chain is enhanced with the trade-off of sharing some useless information among participants of the same shard.

2.2.4 Dimensioning

Some general considerations about the blockchain dimensions: it is possible to lose some transactions, due to overcrowding of participants during peaks in some densely populated cities. But the transactions shared into this network are not the same of cryptocurrency ones, there is no money exchange and the loss of a single transaction does not represent a severe problem, especially bearing in mind that such occurrences happens mostly when the chain is overpopulated, meaning that much information about same targets is being shared. Having fewer transactions addressing the same node, when their number is large, it is not an actual issue. The priority is usually given to fresher transactions since a recent location of a vehicle is more valuable than an old one.

2.2.5 Initial Condition

Few remarks on the initial condition of the framework are analyzed. The starting situation, indeed, is particularly delicate because, without a large number of participants, the network is more exposed. Furthermore, the lack of a consistent database of reputations makes the distribution of first points and the initial growth of reliability an issue to be carefully examined.
RSUs[4] assume a fundamental role. Moreover at the beginning, indeed, when participants have low reputations, it is up to stations, which were granted greater reputations, to identify and be identified in order to launch the growth of the system. RSU transactions are necessary to validate other participants', which in turn will validate stations and other nodes.
Solutions in accordance with smart cities could be taken, such as trusted entities that roam in cities to help initial release.
Several tests have been performed to evaluate how the system responds to different numbers of stations and values of initial reputation granted to new nodes.

Since no existing software was easily adaptable to the framework design,

[4]RSU: Road-Side Unit

a simulator has been implemented in order to meet framework requirements and experiments needs.
The simulator can be parametrized at will and reconstruct particular condition to study system response to possible situation or attacks. Thanks to it, tests were run, and several attack scenarios have been discussed, resulting in mitigations and improvements.
The actual code is further discussed in detail.
Eventually, results and conclusions have been collected.

2.2.6 A brief Use Case

An example, Figure 2.1, shows a common scenario to better understand the algorithm functioning.

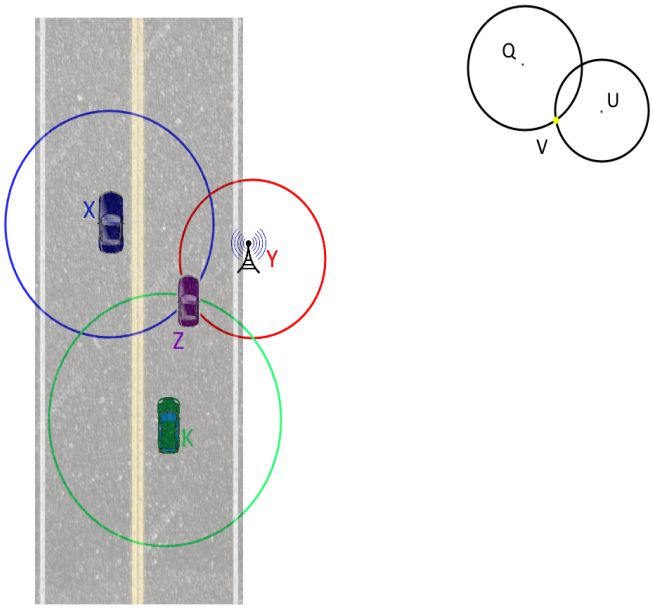


Figure 2.1: The figure shows the situation in a use case, X Z K Q U V are vehicles, Y is a RSU, perception ranges are also showed.

- X Z K Q U V are vehicles contributing to the network update with transactions containing information about reciprocal locations.
- Y is a road side unit station, likewise participating in the network and, by design, more reliable at the beginning.

Assume that participants reputations are defined as in Table 2.1, adapted in this example to values ranging from 0 to 1.

X sends a transaction to all participants of its shard, communicating that Z is 4,5 meters distant.
The same action is performed by Y (3 meters) and K (5 meters).
Elsewhere, at the same time, a similar scenario occurs, where Q states V's location.

Table 2.1: A table with all participants' reputation and their ranges, calculated and overall dedicated.

Participant	Reputation	Dedicated Range	Overall Range
X	0.2	0.0909	0 - 908
Z	0.2	0.0909	90 - 1817
K	0.2	0.0909	1818 - 2726
Y	0.9	0.4091	2727 - 6817
Q	0.5	0.2272	6818 - 9089
U	0.1	0.0455	9090 - 9544
V	0.1	0.0455	9545 - 9999
Tot	2.2	1	0 - 9999

Being all participants under the same shards, and having transactions reached all participants, everyone is aware of transactions generated during the last blocktime: (X $\rightarrow$ Z), (Y $\rightarrow$ Z) and (K $\rightarrow$ Z).
Basing on these transactions, three random numbers are generated, shared among all nodes. Then it is calculated a range for each active participant, driven by their reputation. In this example, the generated numbers are 7,000, 4,000 and 1,000. This means that the election has been won by Q, which will proceed with the creation and validation of the next block, and the support nodes are Y and Z. Q, also, provides for the truthfulness checking of the transactions in the block and communicates who is rewarded or penalized.
Since in this scenario no one shared false information, the transactions are all concordant. The reputation of Y is very high, being it a RSU, while X and K's are average. Q will increase by a small amount the reputation of all the three agents. Otherwise, if K had stated a different distance for Z, or even be in another city, the validator would have realized the false information and severely reduced its reputation.

Eventually, when the whole blockchain is updated, every 15 minutes, a representative for the shard is elected in the same way and participate to an election among all representatives, the winner of this election generate a block with all the blocks from shards and broadcast it to all active nodes, that can now update the state of the whole blockchain, not only their shard.

Chapter 3

Implementation

The implementation focuses on the consensus algorithm and truthfulness assessment, emulating participants' behavior and the transactions exchange among them. Due to the many assumptions that should have been made, and, as previously mentioned, not many data have been produced regarding this topic, implementing an entire system that is composed of different untested parts could have been risky. Without a previous study to ground the project on, many hypotheses might have remained pure guesses.

3.1 System Architecture

The aim of the implemented simulator is to allow the study of a basic application of the framework. It grants the possibility of analyzing participants' behavior and reputation management. The lack of data discouraged the implementation of a real blockchain instance. Nevertheless, its purpose was simulated through a centralized database to which all nodes can access as if each participant were keeping their own common copy of the ledger.

The simulation follows a discrete-time, sending and validating transactions each time step or turn. Time is indeed divided in *turns* that represent the minimum phases in which participants can roam into the map and collect information.
The code is written in JAVA. The implemented classes are Node, Transaction, Position, Dato, CONSTANTS, DrawShapes and StudyCase, which is the main class.

3.1.1 Node

Node represents the entities taking part in the chain. Every instance of this object is rather a vehicle or a station. They randomly roam in the designed map probing for near nodes to identify their position and share the information all over the network. Nodes have access to the database from which they can read or write and are characterized by an ID, a mutable location, a reputation level, a certain range of perception and a Boolean state of activity that describe their condition (i.e. parked and sleeping or awaken on the road). In order to better test the framework, nodes can also be set on malicious, a mode that makes them perform a variety of attacks, which consist in forging particular and sometimes organized transactions.

```
public ArrayList<Node> probe() {
//Simulates the short range connection,
//accessing the whole net..

    ArrayList<Node> nearNodes = new ArrayList<Node>();

    //Probes near nodes other than himself
    for (Node node : StudyCase.participants) {

      if ((node.isActive() &&
        this.currentLocation.distanceToPos(node.currentLocation) <
        this.rfidRange) && node.id != this.id)

          nearNodes.add(node);
    }
    return nearNodes;
}
```

Listing 3.1: The source code of the method probe(), in class Node, to search for near nodes.

Other than its constructor, Node implements the method probe(), Listing 3.1, that simply search, on the whole dataset of current active nodes, for participants that are in the range of their perception; this method was necessary to simulate the proximity communication provided by DSRC. The method generateTransaction(), returns a genuine transaction containing sender node info, target node info, current location and the distance measured between the two nodes. The malicious methods will be further discussed in the experimental validation when attacks are dealt with in detail.

3.1.2 Transaction

Transaction is the class that contains exchanged information. Its composition is simple, made of the same entries discussed in the approach: sender, target, timestamp, location, and distance. For the sake of simplicity, asymmetrical cryptography was not implemented so there is not a signature. Only the constructor was implemented.

3.1.3 Auxiliary Classes

Position represents the coordinates of a location on the map. Before editing one of the coordinates, checks if it is out of the bounds of the map and, in case, bounces back the value of the remaining steps. Dato is just an auxiliary class to better collect outcome results. DrawShapes is a class with many graphic libraries that were used to better understand the data outputs of the tests. A simple map could be printed, with nodes perception range or helpful charts tracking nodes and transactions number. It was used to show results at the end of a test, but also to make charts using as input the long lists of data produced by some days lasting test.

3.1.4 Constants and Parameters

Eventually, CONSTANTS is a fundamental class, it contains all the parameters, making them easily accessible all together.

These parameters are:

- ◇ Map Side dimension
- ◇ Number of Nodes
- ◇ Node speed
- ◇ Nodes perception range
- ◇ Stations number
- ◇ Stations perception range
- ◇ Turns to be simulated
- ◇ Max transactions per block
- ◇ Last blocks to check
- ◇ Minimum transactions with the same target
- ◇ Maximum transactions with the same target
- ◇ Error threshold in distance measuring
- ◇ Starting Reputation
- ◇ Reputation Cap
- ◇ Reward given
- ◇ Penalty given
- ◇ Minimum delta reputations
- ◇ Percentage of Deactivation

Most constants are self-explanatory, but some may not be so immediate to understand.
The `LASTBLOCKSTOCHECK` represents the number of previous blocks to go search through back in the history of chain to gather information when analyzing the truthfulness of a transaction.
The `MINIMUMTALKERS` and `MAXIMUMTALKERS` are the boundaries beyond which a transaction is considered to not have enough agents identifying it, or on the contrary, too many agents and therefore looking suspicious.
`SENSIBILITYDIST` is an epsilon value to allow errors in measurement and still consider an information correct.

`UPREPTHRESHOLD` represents the threshold required above that the sum of reputation of agents stating concordant information must exceed to be considered reliable.
Other parameters concern attacks scenarios and are explained later.

3.2 System Details

StudyCase is the main class, where all the computation is performed. The control flow will be now covered in detail.

Some tests required different parametrization and it was necessary to set the simulator in order to run several tests at once, so there is the possibility to produce different outputs through a single occurrence of the program.

3.2.1 Initialization

First, all the counters, central database and variables used to measure data are initialized. Nodes are now generated and only a part of them are set awake, ready to roam, as a more realistic scenario. A percentage of the participants is randomly set to act following a malicious policy. Stations are also added to the map. Every node is randomly placed into the map. The borders of the area are set for simplicity to be impassable, nodes that encounter them just bounce back and keep following their track. This is not a restriction of generality because nodes can activate and deactivate themselves in each time step, being non detectable as if they were out of the map.

3.2.2 Collection phase

Then the major loop, representing the discreet time, starts. Each step of discrete time is a turn, that would represent a minute in real scenario. In

this phase, participants first roam into the map, then at the end, examine the shared transactions.
In every step of the loop, all active participants perform a random step in a direction, for test purposes the step is only one and having a length between a positive and negative value representing the distance covered at the maximum speed chosen. Each node probes its proximity and if a near node is detected, it generates a transaction with the correct information and sends it to the distributed ledger, which is replaced by a central database in this simulator.
Then different attack scenarios can be activated, and only malicious nodes perform these steps. These actions are analyzed in details in Section 4.6 but, in general, attackers always try to trick the system sharing false transactions with the purpose of obtaining undeserved reputation or infecting the chain with wrong locations.
Each turn, every node has a chance to be switched off or get awaken.
At the end of the discreet turn, there is the phase of validation that is performed as a sort of election, to draw the next validator and the two support nodes. The election is implemented following the guidelines exposed in Chapter 2, reputations of all active participants are combined side by side in an ArrayList, three distinct numbers determine the validator and the supports. In the case of this simulator, the election is useful to study percentages of wins pursued by malicious nodes because they will raise all nodes' reputation regardless.

3.2.3 Evaluating Transactions

Follows the verification of shared transactions, if they exceeded the `BLOCKLIMIT`, they are randomly sorted and the first `BLOCKLIMIT` transactions are picked. In the actual implementation, this step must take into consideration randomness that can be reproduced by support nodes too, such as selecting which transaction to evaluate according to the ID of the validator.
For every transaction, it is collected a group of other transactions, coming from the same block and from the past `LASTBLOCKSTOCHECK` blocks, that

have the same target of the analyzed one. These are the *talkers*, the fresher ones are more relevant, indeed their reputation is multiplied by a decreasing value, ranging from 1, for the current block, to 0 for all blocks older than `LASTBLOCKSTOCHECK`. If the analyzed transaction shares the target with at least `MINIMUMTALKERS` other transactions, likewise weighted according to their freshness, then its truthfulness is evaluated.

```
if (x > CONSTANTS.UPREPTHRESHOLD ||
    (CONSTANTS.ATTACK && CONSTANTS.ATTEXTRACTION && attackerWon)) {

    StudyCase.reward(trans.getSender());
    totalRepChanges++;

    if (trans.getSender().isMalicious() && t > CONSTANTS.TURNSBEFOREATT)
        falsePositives++;}

else if (x < -CONSTANTS.UPREPTHRESHOLD) {

    StudyCase.punish(trans.getSender());
    totalRepChanges++;

    if (!trans.getSender().isMalicious() && t > CONSTANTS.TURNSBEFOREATT){
        falseNegatives++;
    }
}
//otherwise do nothing
```

Listing 3.2: The source of reward or punishment of an analyzed transaction. Also, checks if a malicious node is drawn, rewarded or punished.

Now, as discussed in Chapter 2, talkers are divided into two groups, concordant or discordant, observing the formula 2.2.3. If the majority of reputations sum leans to the concordant group, the sender is rewarded, otherwise, it is penalized. To assist this process, some mitigations come into play in this phase to reduce error or malicious intents. These restrictions are formalized as result of the outcomes of the experiments analyzed in the following Chapter 4.
Some of these methods, however, lead only to a slight mitigation of attackers' trend. One example is the `MINREPEATPAUSE` parameter. Two vehicles are unlikely to be near for extended periods of time but rather when this happens in a real scenario it may represent an attempt, by malicious participants, to obtain reciprocal reputation. This is why a node cannot gain further reputation, i.e. transactions are not considered, when they submit information regarding the same node once every `MINREPEATPAUSE`

minutes. This can reduce the possibility of improper reputation gain, but on the long run it is not a suitable solution.

3.2.4 Accessory Code

The turn ends and the loop starts back, all the code that comes after is for statistic and result purposes. There are several reputations calculations, analysis of percentages of genuine and malicious nodes and the relevance they had in the system.
There is the possibility of analyze shared transactions during the simulation, which is useful to calculate how many transactions were generated by malicious nodes toward a specific target, in comparison to those generated by genuine ones. These data are used to calculate the percentage of false information the system can deal with. Execution time is measured too.
There is also the option of saving outputs and information into a file, to be later introduced as input for chart production.

Chapter 4

Experimental Validation

A crucial part of the research is represented by the tests and experiments run to simulate the environment. To collect these result we made use of the implemented simulator to foresee the behavior of framework.
The following sections describe each test typology, alongside the reasons and goals for such test, the difficulties encountered and how we overcome them. Will be clarified how the population number and map size were obtained and justifications for the adopted dimensions.

Description of the experimental setup that has been adopted and computer specifications are also reported. Some charts and pictures will be provided in order to ease the understanding of experiments.

4.1 Goals of the tests

The main goal of all the tests is to analyze the feasibility of the framework. These experiments have been fundamental due to the lack of concrete examples; the system, in fact, could have shown unexpected behaviors when facing corner cases, that may have been hard to foresee without testing.

Also, we wanted to test the framework against known attacks that could be faced in the wild. After analyzing possible attackers' mindset and objectives, different types of attack have been performed.

4.1.1 Type I - Feasibility Study

The first type of performed test aimed to evaluate feasibility, scalability and possible divergences of the system during its ideal functioning. Participants during these tests were following a genuine behavior, they all had the same perception range and potentially the same transactions throughput.

The dimension of the map and the number of nodes were tailored according to parts of the city of Milan, although in some experiments the numbers were downsized, while keeping the same density of nodes, to maintain acceptable test times. Different initial conditions were analyzed, also varying the number of stations, density, and reputations of participants.

4.1.2 Type II - Attack Resistance

The purpose of the second typology of experiments is to test the response of the system when it is subjected to different threats. In these scenarios, various percentages of nodes were designed to act as malicious agents: they could forge fake transactions to state the victim's location somewhere else or helping other attackers to grow each other reputations. Attackers could also behave normally to obtain a consistent reputation and then strike an organized attack toward a common target. Parameters that could be set during these experiments are the percentage of attackers, the possibility to activate each attack mode separately, the turns to wait before letting the attack begin, the targets of a distributed attack and the ratio of attackers' computational power respect to normal users.

Tests have been useful to better understand system robustness, evaluate limits and security threshold above which the framework is endangered. The percentage of attackers it can handle and the ratio of genuine and malicious transactions toward a single target can be managed. Eventually, we evaluated false positives and negatives for each test, considering as a false positive the event of a malicious node being rewarded and as a false negative a good node being punished.

4.1.3 Challenges Encountered

The most challenging part has been the research of precise data to adjust simulator's parameters. Indeed, it was not easy to find solid data about the proper road network of Milan or another big city. The same issue has been encountered when searching for data on vehicles circulating daily since available information was too vague to be effectively useful.

Often statistics were obsolete or without a proper source, sometimes the release dates of studies on viability and road information were too distant from one another to be considered alongside. Thus, we applied some approximations. To ensure the feasibility of the system, these approximations were always oversized so that if tests are successful with a more fragile sizing, the actual framework is indeed more robust.

4.2 Dataset and dimensioning

The feasibility tests were dimensioned in order to respect, as realistically as possible, the territory and areas in which cars can roam within the borders of the city of Milan. Since participants of the simulator freely roam into the map, dimensioning an area as large as the surface of Milan, without considering streets distribution, would have not been accurate.

That is why we generated a map consisting only of road surfaces. Data taken from the official site of Milan Municipal [37] show how the urban area of Milan is covered about of 1947 km of streets, of which 48 km are roads of the highway network, 359 km are represented by neighborhood streets and 1540 km of local roads[1].

Once obtained an approximate length of all the road considered, excluding highways, thanks to the study [38] that reported how the average width of Milan streets is 5.02 meters and the ratio of roads distribution is 1.47 km per km^2 of urban area, it has been possible to calculate an approximate road surface on the Urban area of the city.
Having the test area is not sufficient to run experiments without an appropriate number of participants; however, a previous research [39] has been

[1]Respectively, according to Italian road regulation, strade di scorrimento, strade urbane di quartiere and strade locali, art 2 codice della strada

useful to esteem this number.

The research produced the following results:

- ◇ **Cerchia Bastioni** area, known as Area C, that is about 8.138 km^2, registers daily 158k accesses.
- ◇ **Cerchia Filoviaria** area, stretching for 28.728 km^2, registers about 423k daily accesses.
- ◇ City limits, the **municipal boundaries** of the city, that is basically the Area B, 181.8 km^2 large, counts 630k daily accesses.
- ◇ Eventually, the whole **Milan Province**, large 1575 km^2, has a less accurate number of about 1.7 - 2 Million daily accesses.

On the other hand, attack experiments were run following other settings. Considering that performing tests in an ideal scenario would not have provided an indicative robustness index, these tests adopted harsher conditions.
One of the best defenses of blockchain technology lies in a high throughput of correct transactions, against which no attacker has enough computational power to drive it on their behalf. This means that if a system can resist an attack performed by a certain percentage of malicious agents, it will also advisedly resist if the number of genuine nodes increases.

This is why the dimension scaled in favor of attackers, boosting their percentage in relation to overall nodes. Attack scenarios indeed, will come with, not only a greater percentage of malicious nodes, but also a smaller density of transactions, making the system less robust to organized attacks for testing purposes, also reaching collapse threshold, breakdown, and overall failure.

4.3 Experimental Setup

The machine used to run tests was hosted on one with the following characteristics.
Dell PowerEdge R720xd, two processors Intel(R) Xeon(R) CPU E5-2680

v2 @ 2.80GHz, 10 cores each for a total of 20 cores, with hyperthreading enabled, 377 Gb RAM. The Virtual Machine actually used to run tests was set with 4 virtual CPUs, 16 GiB RAM, and 40 GiB HDD.

Tests did not rely on the performance of the machine they were run on. No computational time biased the results, nor available memory.
The code can be executed on any kind of system, as far as it is able to compute the code, and the produced result will be much similar; nevertheless, it should be kept mind that many variables are conditioned by random values, making exactly the same data difficult to obtain again.

4.4 Feasibility Tests

4.4.1 Experiment 1: General Simulation

This simulation analyzes the behavior of the framework under an ideal, normal, condition, with no attacks. The instance should resemble the city of Milan on usual conditions. The covered area is the scale of $\frac{1}{4}$ of Municipal Boundaries of Milan, reminding that this area is not the totality of the territory, but an approximation of the road surface in the area. The map is a square with the side of 1500 meters, resulting in 2,250,000 m^2. In this area are roaming 107,500 nodes that are normal users and 250 high-reliable stations. This results in a density of 0.05 vehicles per m^2, dedicating to each node 20 m^2, but having a perception range of 314 m^2. The required reputation for a transaction to be considered is 75 and at least 2 *talkers* are necessary. 10 discrete turns are simulated.
The outcome provides the following information:

```
Average transactions per turn:  394292
Total average reputation:  67.996750
Average Reputation of normal users:  66.045395
False positives:  0    0.000000%
False negatives:  0    0.000000%
Total reputation changes:  3914225
Total transactions:  3942928
Execution time 31 h 35 min 32 sec
```

Figure 4.1 is a simple chart showing reputation growth during the 10 turns of ideal framework behavior. The results were quite expected since no attack policy has been adopted, the transaction throughput is high but lower than the one that the proposed blockchain, and the nowadays working IOST's, can manage. It should be noticed how the average reputation grows linearly with a very high coefficient; this is due to the extremely high density of participants, a transactions per block limit can be introduced to lower it. In the following tests will be observed how that growth will not be so swift.

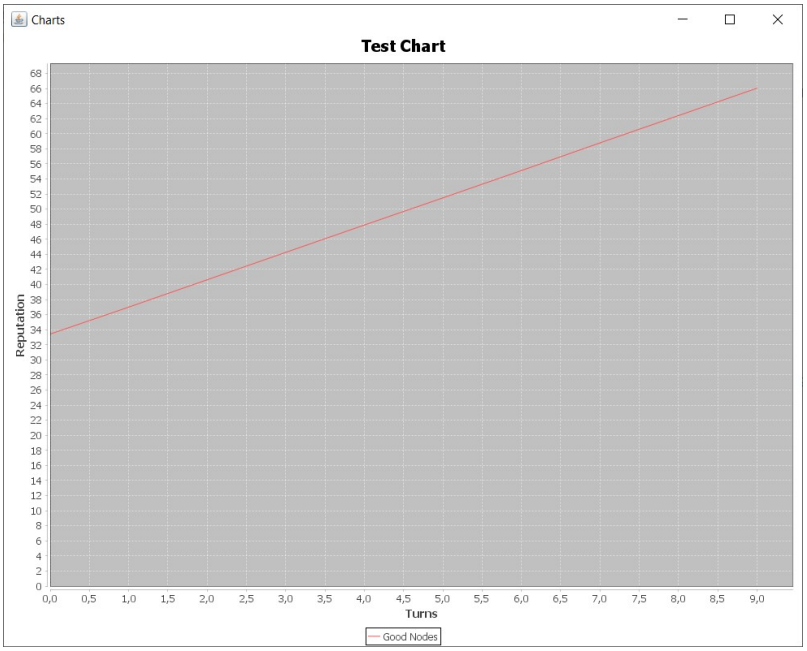


Figure 4.1: A chart representing the variation of average reputation in relation to turns shifting.

4.4.2 Experiment 2: Starting Reputations

The second experiment dealt with the study of the system response to different values of reputation given to new users. Starting trust is indeed an important property to decide how to approach to new users. In all the tests, the map was regulated to be a 2 km side square, with 10000 participants and 250 high-reliable stations. 100 turns were studied. The 20% of nodes is dynamically deactivated. Distinct policies were adopted. The first is a safer one: be wary of new users, considering them unreliable, giving them absolutely zero trust; but, at the same time, granting them the opportunity of committing transactions to earn reputation points, when the information they share is carried on by other, reliable, users too. This policy requires high-trust agents on the first, to allow the framework to catch on.

The output is the following, Figure 4.2:

```
Avg transactions per turn:  5487
Average total reputation:  60.647221
Average nodes reputation:  38.437302
Total reputation changes:  434143
Total transactions:  548783
```

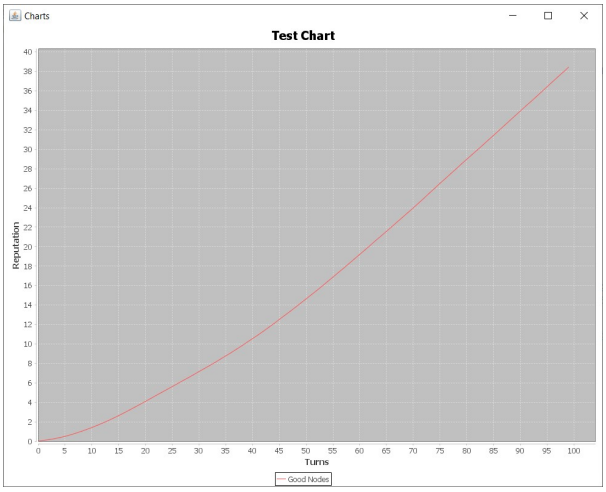
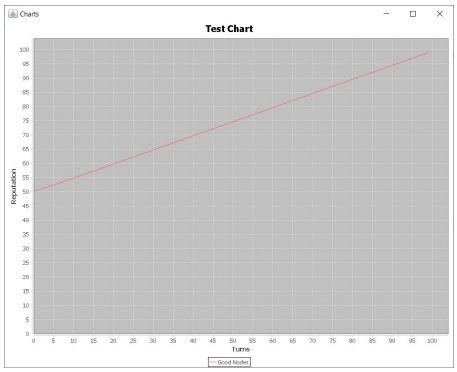
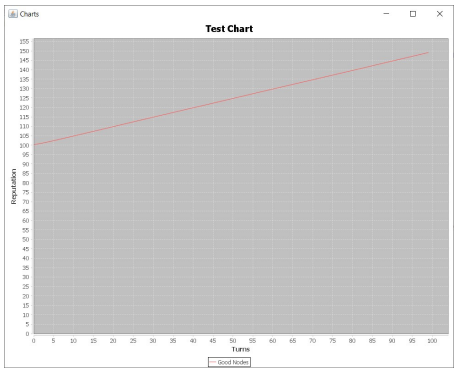


Figure 4.2: A chart representing the variation of average reputation in relation to turns shifting. With starting reputation of 0.

Other experiments tested higher reputations value at the beginning, in particular, 50, 100, 500. Outcomes and charts are below, after which a brief comment on results will follow.



(a) `Avg transactions per turn: 5468`
`Average total reputation: 119.671806`
`Average nodes reputation: 98.946297`
`Total reputation changes: 539160`
`Total transactions: 546884`



(b) `Avg transactions per turn: 5494`
`Average total reputation: 168.716873`
`Average nodes reputation: 149.149399`
`Total reputation changes: 541848`
`Total transactions: 549434`

Starting Reputation at 50, chart in Figure 4.3a

Starting Reputation at 100, chart in Figure 4.3b

Starting Reputation at 500, chart in Figure 4.4

Eventually, a comparison chart with a recap of all trends in Figure 4.5

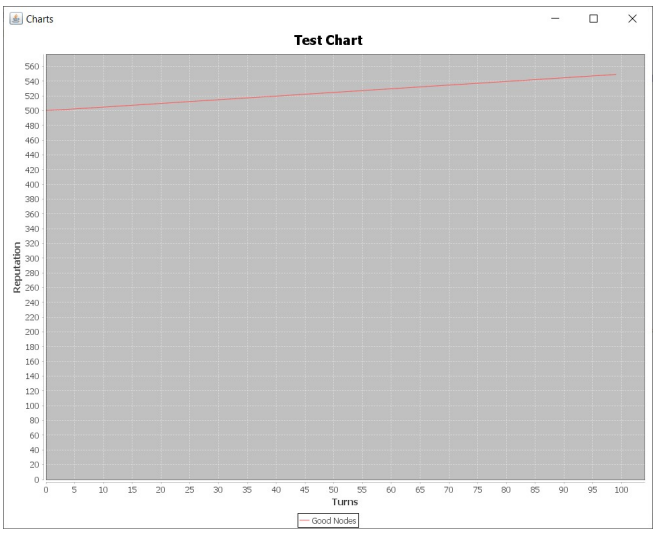


Figure 4.4:
```
Avg transactions per turn:  5483
Average total reputation:  558.835327
Average nodes reputation:  549.041016
Total reputation changes:  540566
Total transactions:  548324
```

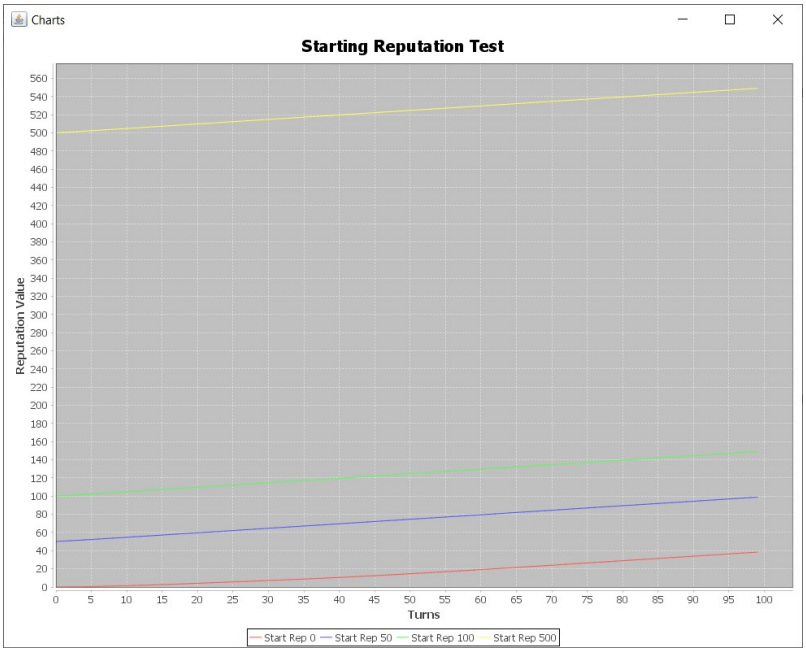


Figure 4.5: A chart representing the different values of reputations growth in accordance with various values of starting points.

Thanks to the comparison of the outcomes in Figure 4.5, it can be noticed how, predictably, the average of reputation changes, but the same did not happen to its growth rate. As far as nodes have a consistent reputation, they will easily exceed the `UPTHRESHOLD`, a parameter that requires a minimum sum of talkers' reputation in order to consider a transaction valid, and will be discussed deeper in the next test.

Eventually, checking the outcomes, the total amount of *Total reputation changes*, that is the number of times a node has been subject to an update of its reliability, and *Total transactions* are correlated. The latter number is the total number of transactions that were analyzed, while the former represents only those that caused a reputation update.
If they had been equal, it would have meant that every transaction committed caused an update in the sender's reputation. If the number is much lower, it is because many transactions were discarded due to the low reputation level of their talkers, has happened in the first case of test.

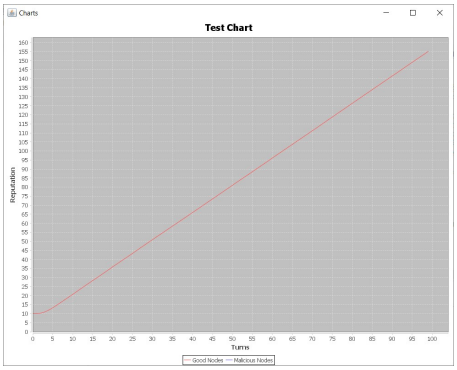
4.4.3 Experiment 3: Trusted Entities Dependency

In the previous experiment 4.4.2, it was noticed how in some scenarios, high-reliability entities are necessary to let the framework catch on.

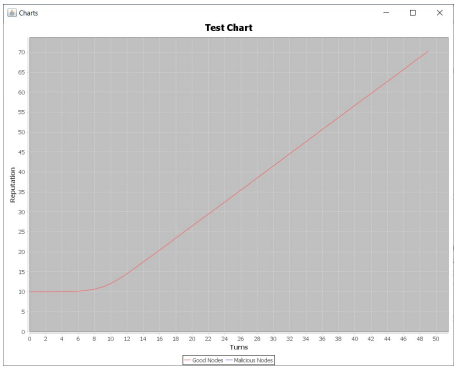
The following two tests show the initial behavior when the ration between initial reputation and `UPTHRESHOLD` is respectively 0.133 and 0.067; meaning that the minimum number of vehicles to meet in a minute to evaluate those transactions must be at least 7.5 and 15 when considering the initial situation.

The input is 1000 meters side, 5000 nodes, 100 turns.

First test	Second test
Initial reputation: 10	Initial reputation: 10
Up threshold: 75	Up threshold: 150
Figure 4.6a	Figure 4.6b



(a) A chart showing the starting condition of the framework when there are no high-reliability stations, and the threshold to accept a transaction is 75,initial 10.



(b) A chart showing the starting condition of the framework when there are no high-reliability stations, and the threshold to accept a transaction is 150, initial 10.

We now propose a comparative Chart 4.7 showing different ratios of reputations and uptresholds; in all cases, participants start with an initial reputation of 50. In the first two scenarios, the absence of stations has caused nothing but a slowdown of the entry into full operating phase, the same happens in the first 3 cases in Figure 4.7. When the threshold is 1200, having thus a ratio of 0.041, the system has difficult to start behaving correctly but after a while it does.
In the last two cases, instead, the requirements are too harsh, relating to the initial condition, to allow the development of the framework; these conditions, indeed, require 50 vehicles to perceive the same target in a minute.

Table 4.1: A table showing the parameters which outcomes are showed in Chart 4.7. Starting reputation is 50 for every case.

UpThreshold	Ratio	Talkers required with initial reputation	Turns before catch on
400	0.1250	8	3
600	0.0833	12	5
800	0.0625	16	9
1200	0.041	24	80 - 130
2000	0.025	40	did not occur before 2000
2500	0.020	50	did not occur before 4000

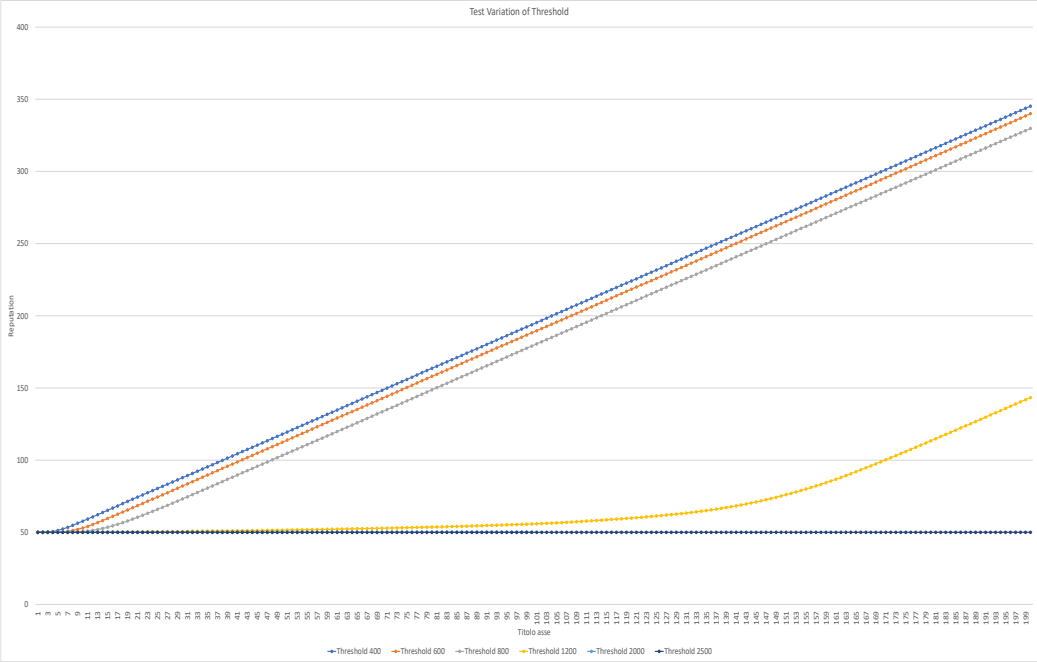


Figure 4.7: A chart showing the comparison between different ratios of initial and required reputation to consider transactions. The two lower ratios could not let the system function properly.
Ratios are in order: 0.1250, 0.0833, 0.0625, 0.041, 0.025, 0.02.

4.4.4 Experiment 4: Node Density

The purpose of this category of experiments is to evaluate the framework functioning under various conditions of participants' density. The whole system is based on transactions and having a consistent number of them circulating in the blockchain is fundamental. Of course, transactions throughput is strongly correlated to the number of users, but mostly to their density; indeed, it is not sufficient to have a large number of users if they cannot perceive each other.

The outputs will measure the reputation changes caused by nodes encounters and simple pictures of the population. The images represent the perception range of every node, they are useful to get an immediate visual feedback of the scenario, to better understand experiment outcomes.

The input of the first test includes a 1000 meters side square map, 5500 nodes, no stations, a starting reputation on 50 for all participants. The density in this scenario is 0.0055 nodes per m^2, keep in mind that the perception range of nodes is still 314 m^2 (10 meters of DSRC range). Figure 4.8a and 4.8b.

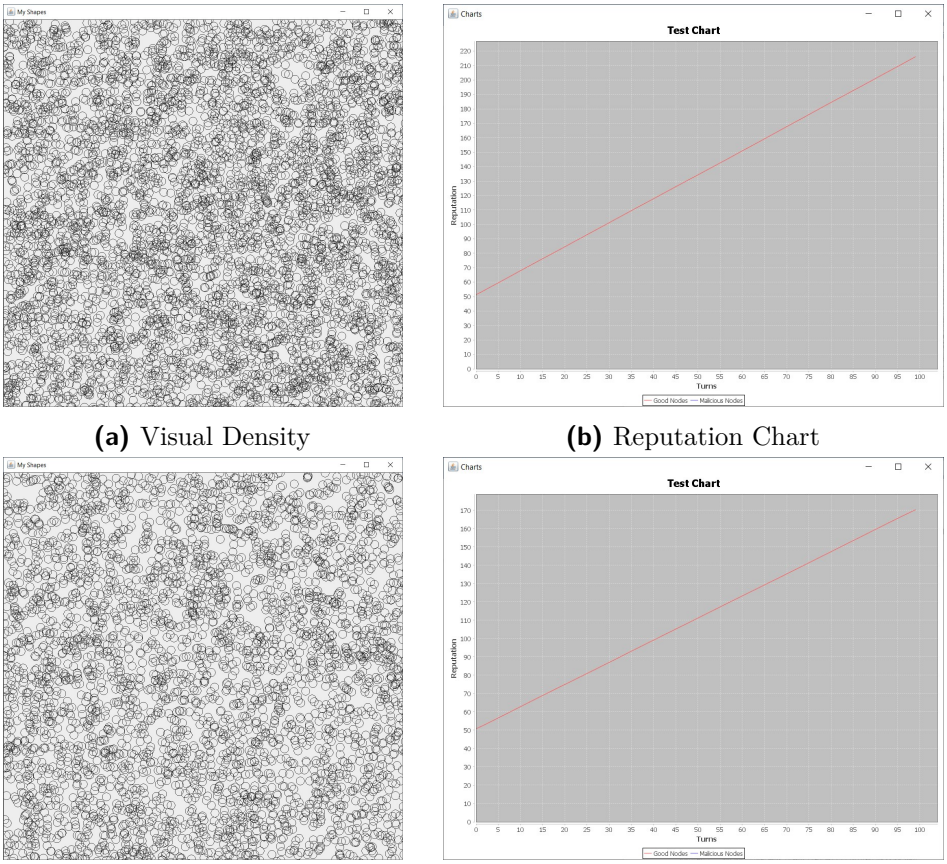
Second experiment, the input is equal, just changes the number of the nodes and thus the density. 4000 nodes, results in 0.004 nodes per m^2. Figure 4.8c and 4.8d.

Third test, 3000 nodes, density of 0.003 nodes per m^2. Figure 4.9a and 4.9b.

Fourth test, 2000 nodes, density of 0.002 nodes per m^2. Figure 4.9c and 4.9d.

Fifth test, 1000 nodes, density of 0.001 nodes per m^2.

Nodes encounters start getting very rare, 500 nodes, density of 0.0005 per m^2. Figure 4.10a and 4.10b.



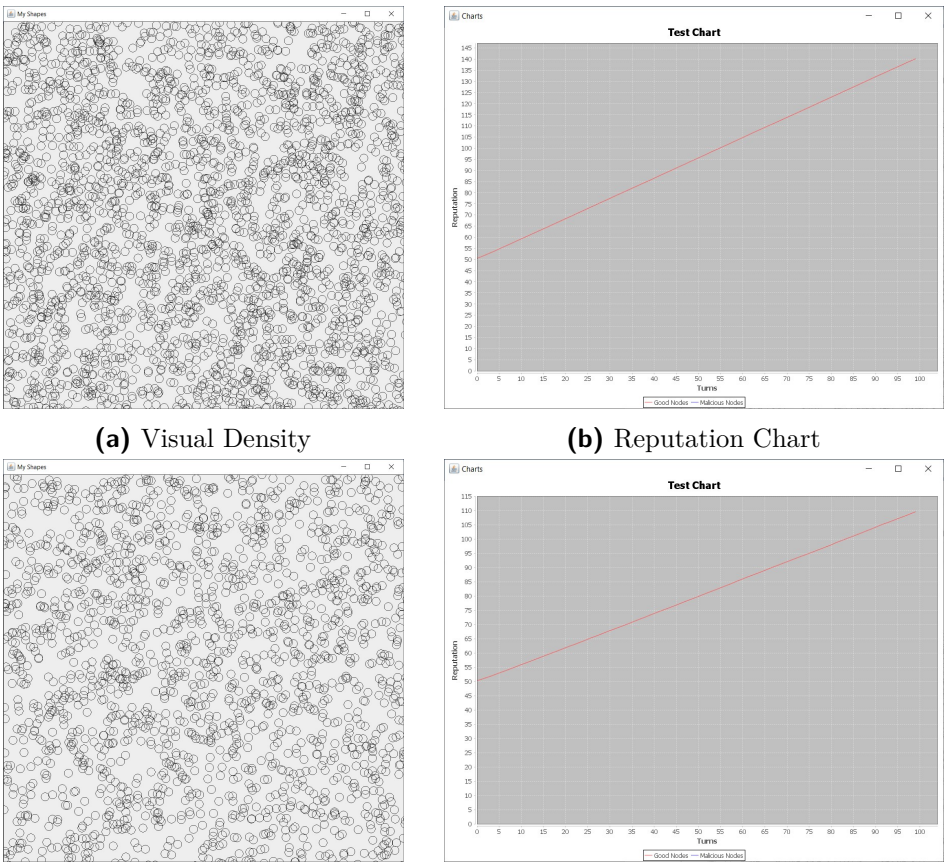
(a) Visual Density

(b) Reputation Chart

(c) Visual Density

(d) Reputation Chart

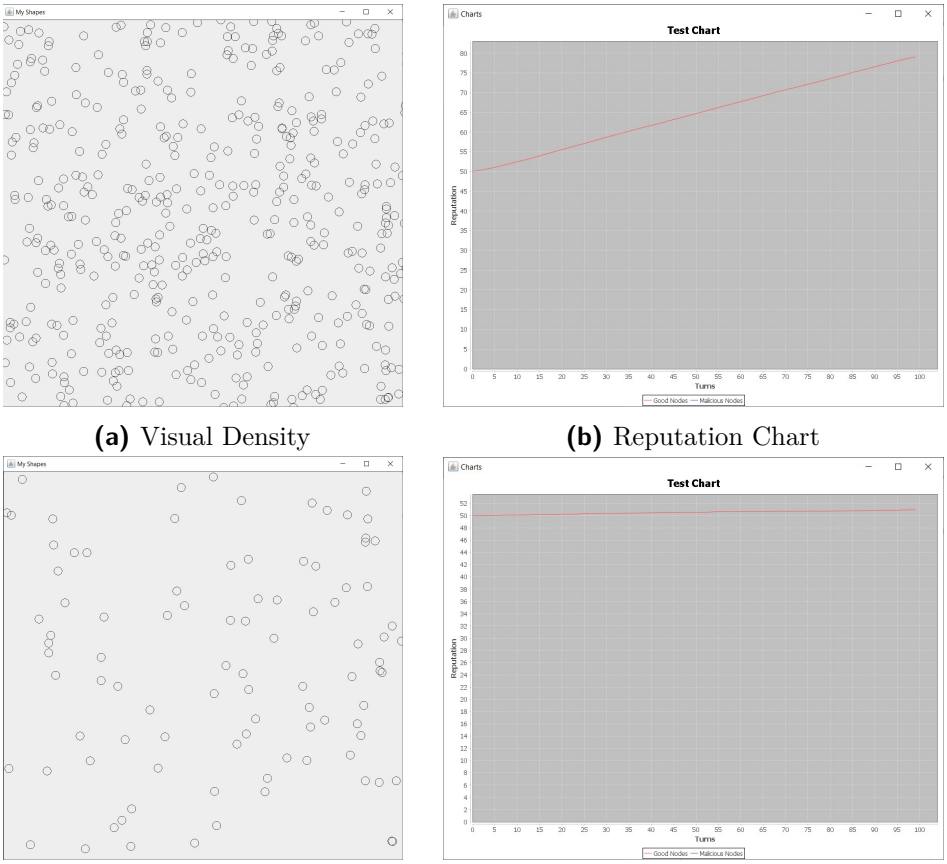
Figure 4.8: Nodes distribution and reputation changes with a population density of 0.0055 nodes per m^2 in Figures (a) and (b). Density 0.004 nodes per m^2 in Figures (c) and (d).



(a) Visual Density **(b)** Reputation Chart

(c) Visual Density **(d)** Reputation Chart

Figure 4.9: Nodes distribution and reputation changes with a population density of 0.003 nodes per m^2 in Figures (a) and (b). Density 0.001 nodes per m^2 in Figures (c) and (d).



(a) Visual Density **(b)** Reputation Chart

(c) Visual Density **(d)** Reputation Chart

Figure 4.10: Nodes distribution and reputation changes with a population density of 0.0005 nodes per m^2 in Figures (a) and (b). Density 0.0001 nodes per m^2 in Figures (c) and (d).

Eventually, almost no transaction at all is generated when the nodes are 100, and the density is 0.0001 nodes per m^2. It is graphically evident how nodes are not likely to meet in this condition. Figure 4.10c and 4.10d.

Table 4.2 shows a summary of previous situations, highlighting the number of shared transactions and the reputation updates per turn. The system seems to be stable and be self-sustained when density is above 0.0005 nodes per m^2, meaning that there should be at least an average of one vehicle every 2000 m^2.

Table 4.2: A table showing behavior of the framework in accordance with different population density conditions

Nodes Number	Density	m^2 per vehicle	Transactions per Turn	Δ Reputation	Increment per Turn
5500	0.0055	181	9150	165	1.6
4000	0.004	250	4835	120	1.2
3000	0.003	333	2724	85	0.8
2000	0.002	500	1205	60	0.6
1000	0.001	1000	303	29	0.3
500	0.0005	2000	77	13	0.1
100	0.0001	10000	2	1	0
50	0.00005	20000	0	0	0

4.4.5 Experiment 5: Evolution of Transactions Quantity

The last feasibility test studies the correlation between population number and the quantity of transactions exchanged. Hundreds of tests were run, all with same input, but a varying number of nodes; the total amount of transactions represented the outcome.
The first tests feature a population increment of one for each test, they are more precise, but could have been performed only on overall lower populations.
Figures 4.11, 4.12a, 4.12b and 4.13a show the interesting exponential relation between nodes number and exchanged transactions, this relation is more accentuated when comparing low population values.

The chart in Figure 4.13b demonstrates how the relationship is maintained also when dealing with greater numbers, the step is now an increment of 20 nodes each.

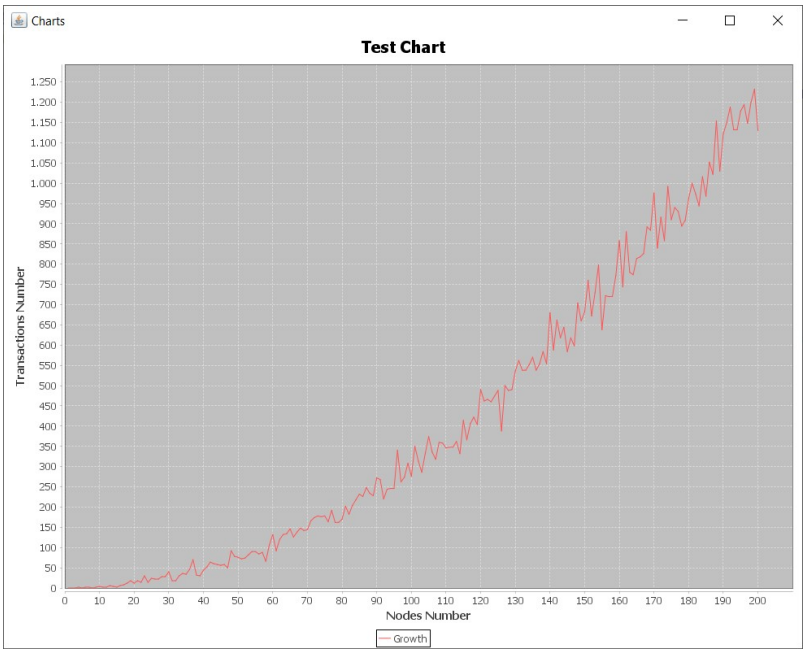
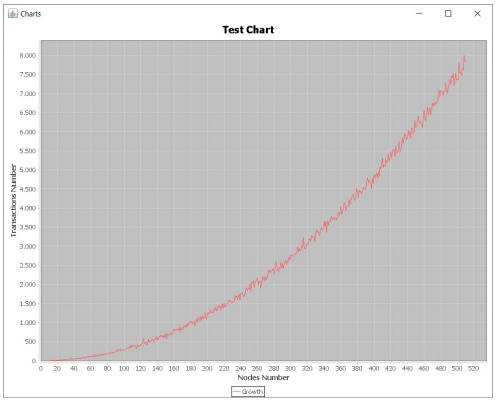
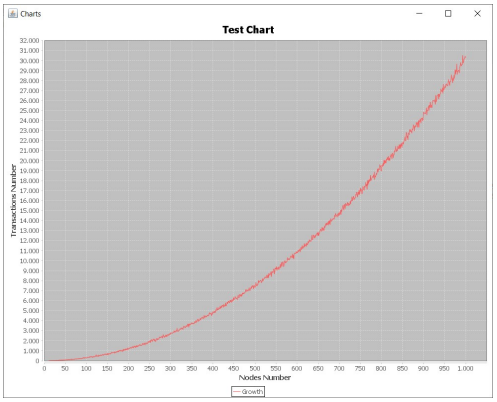


Figure 4.11: A chart showing transactions throughput varies with an increment of population. Variation 0 - 200 showed here.



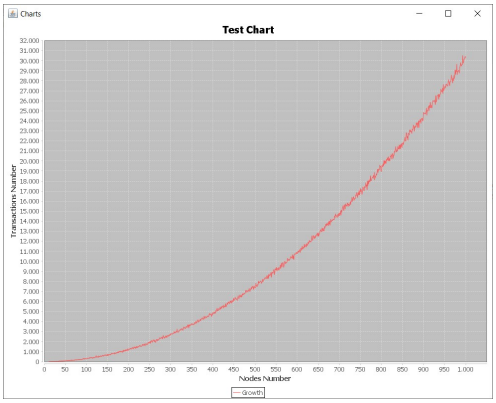
(a) A chart showing transactions throughput varies with an increment of population. Variation 10 - 510 showed here.



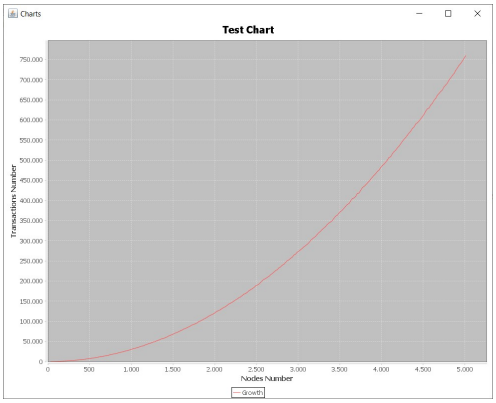
(b) A chart showing transactions throughput varies with an increment of population. Variation 500 - 1000 showed here.

Figure 4.12

The consistency of the exponential relationship is remarked in the chart in Figure 4.14 that shows how the trend is retained.



(a) A chart showing transactions throughput varies with an increment of population. Variation 10 - 1000 showed here.



(b) A chart showing transactions throughput varies with an increment of 20 nodes each step. Variation 30 - 5010 showed here.

Figure 4.13

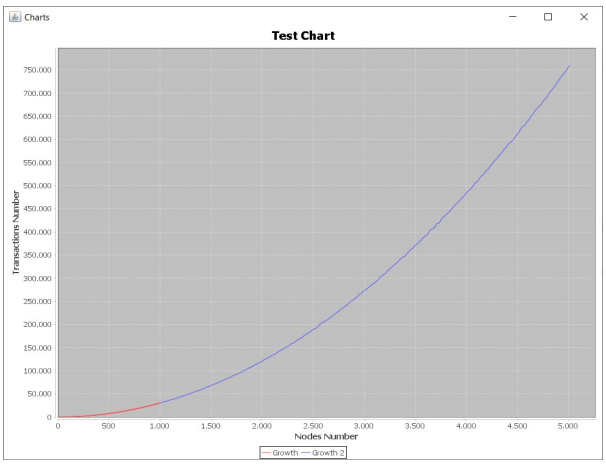


Figure 4.14: This is a comparison between the trends of 10 - 1000 variation, and 30-5000. The trend is similar.

Eventually, an interesting test was run to study the different responses in transactions number when high-reputation wider-ranged stations are introduced in the system. Figure 4.15. This last experiment shows how the introduction of stations surely increases the number of transactions, due to their obvious contribute increasing population number, and also their wider range to perceive more vehicles, but also makes clear how they are not mandatory once the system has caught on its full operational functioning.

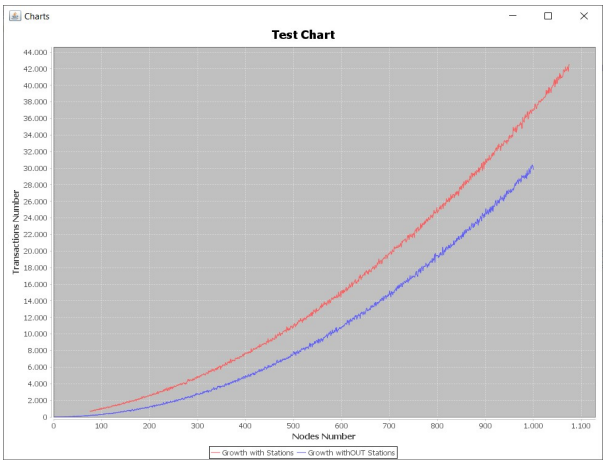


Figure 4.15: The figure shows the comparison in the trends generated by a scenario with no stations and one with 75 of them.
Trends 0 - 1000

4.5 Threat Model

The system is by design open to anyone. Any participant can send a message into the blockchain because of it being open to everyone. This property is a great quality that allows decentralization, but also, exposes the framework to several types of attack. The system is self-maintained, meaning that there are not moderators or entities that can undertake modification with absolute control; neither can participants be estranged by default.
The algorithm proposed can reward or reduce the reputation of a participant according on their statements on the chain, however, its adoption may be a standard, but not a mandatory regulation.

4.5.1 Attackers

Corporations that could be disadvantaged by this project may try to hinder its grown. Hackers or groups of fraudulent professionals could be hired to damage the network. Expert people may just want to test how robust is the system for study, performing various attacks.

4.5.2 Purposes

The purpose of attackers can vary. It can range from a classic denial of service, aiming to disrupt the framework and cutting out people from the service or, in worse scenarios, injecting malicious transactions to let the system work, but not as it is supposed to, creating false information and confusion.
Attackers may even try to guide the traffic of an entire city, even though this would require an enormous effort, that could be pursued only by an organized group of specialists.

4.5.3 Risks

As in the majority of networks, a DDoS[2] attack has an extremely high rate of success, given that the attacker has enough resources to flood the network. in this case, it would require a massive number of participants since the number of accepted messages per node can dynamically change and, in the case of noisy participants, they can be regulated.
The mislead of town traffic can be achieved if an attacker organized it in advance. Adopting a genuine behavior to obtain reputation over a long span of time, allows attackers to be reliable when they strike altogether, but their influence, if successful, would last for at most a minute, and all the malicious node will be punished after. This is not a win situation for the attacker, but still, should be considered.
The second, most important set of threats we proceed to discuss is that of injection attacks along with all those attacks that aim at inserting malicious transactions as valid ones.
Also, attacks may lead to a generalized credibility loss in the system, that would result in a reduction of the number of participants which, in turn, makes the whole system less secure.
Finally, injected transactions can lead to the system being less reliable due to the reputation loss of genuine nodes.

4.5.4 Vulnerabilities and Countermeasures

One of the most exposed aspects of the blockchain is its property of being open to everyone, that can represent a huge vulnerability if exploited properly. Every participant has the same right to share information, which is both a vulnerability and a feature.
In many blockchains, this represents one of the most significant issues, attackers generate plenty of fake users, all on their command, so to be able to drive information as they want, thanks to multiple concordant transactions [40].
One of the first countermeasures to deny the possibility for an attacker

[2]DDoS: Distributed d Denial-of-Service

to control many nodes is to preserve the identity of participants ensuring consistent key management. Private and public keys, indeed, allow users to commit transactions assuring their identity, and without them, malicious agents could not pursue their intent. We proposed a couple of potential solutions in Section 2.2.2, entitling whether car dealers or Department of Motor Vehicles to distribute officially the pair of keys to users, so to bound the generation of transactions to an individual vehicle or driving license. Whichever is the most suitable way to reach this goal, however, is left to further studies since this is not the scope of this research.
Even if keys were bounded to driving licenses, participants' identity will be preserved since blockchain entries are made with ID and a digital signature, resulting in a sort of pseudonymity.

The framework does not have a central, vulnerable and attackable server; this makes necessary an attack to be distributed as well and target several nodes.
The main framework defense relies on the number of participants. Being themselves the heart of the blockchain, the more agents make use of it, the safer and more difficult to attack it will be. Theoretically, with a huge number of genuine users, the structure is essentially safe.
Anyway, some mitigations have been proposed to cope with general threats.

4.6 Attack Tests

To simulate the attacks, as previously explained, some nodes will adopt malicious policies. They will still behave correctly but also, during the turn, they will perform malicious actions. In the simulator, attacks can be individually activated and parametrized at will.

4.6.1 Experiment 6: Random Attack

This is an attack that can be pursued by singular unorganized participants. The attack has been defined "Random" because the targets are picked without a structure; each attacker acts on their own, aiming to a

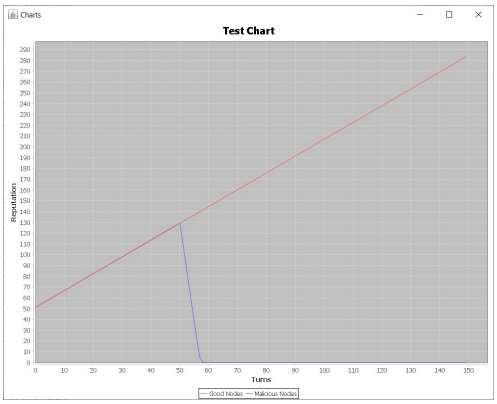
certain target, communicating the same fake position every turn. This may both occur due to some kind of malfunctioning or on purpose.

The input of the test was defined as 1000 meters side square map, 5000 nodes, and 150 stations. The percentage of attackers is 15%, converting into malicious nodes the participants among those 5000. Attackers start acting maliciously after 50 turns of normal behavior. 150 turns are simulated, and the reputation of new nodes starts at 50.

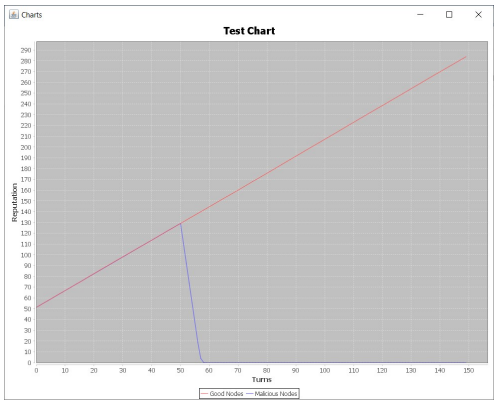
The outcome is the following:

```
Average transactions per turn:  9233
Number of genuine nodes:  4250
Number of malicious nodes:  750
Average reputation of all participants:  263.075531
Average reputation of genuine nodes:  283.491516
Average reputation of malicious nodes:  0.000000
Total Reputation Changes:  1383100
Total transactions:  1385017
```

Chart in Figure 4.16a



(a) The chart shows the average reputation of normal nodes and malicious nodes. 15% of malicious nodes. The attack is unsuccessful.



(b) The chart shows the average reputation of normal nodes and malicious nodes. 75% of malicious nodes. The attack is unsuccessful.

Figure 4.16

A similar attack was performed, but this time the independent attackers represent the 75% of population.

The output is analogous:

```
Average transactions per turn:  11229
Number of genuine nodes:  1250
Number of malicious nodes:  3750
Average reputation of all participants:  98.007576
Average reputation of genuine nodes:  283.79119
Average reputation of malicious nodes:  0.000000
Total Reputation Changes:  1682381
Total transactions:  1684417
```

Chart in Figure 4.16b

Both the attacks fail in their intent of creating confusion in the ledger. This is because the locations declared by malicious nodes regarding victims are unrealistic and much farther from actual locations, so they are easily spotted as incorrect.

If we try to make attackers claim a possible location, near the actual one, for the victim to be, the result will be much different since there is not a precise manner to understand the direction a vehicle has taken after encountering another agent. The validity check, indeed, measures the diametrical opposite possible directions in order to evaluate all the possibilities.

This test shows how stating a more credible fake location, might actually generate confusion.

```
Number of genuine nodes:  4250
Number of malicious nodes:  750
Average reputation of all participants:  317.573212
Average reputation of genuine nodes:  283.270813
Average reputation of malicious nodes:  375.467987
Total Reputation Changes:  1380869
Total transactions:  1382926
```

Chart in Figure 4.17

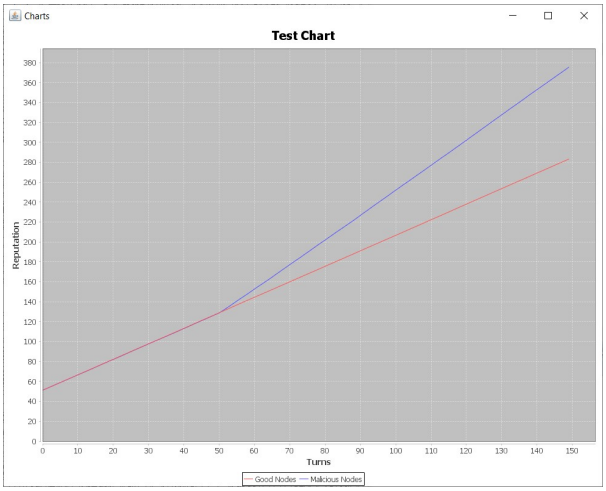


Figure 4.17: The chart shows the average reputation of normal nodes and malicious nodes. 15% of malicious nodes. The forged location is a possible one, the attack succeeds.

The attack was successful because setting a high maximum speed of nodes in a smaller map than the real one, after a few time steps, the information stated by malicious nodes will be actually possible.
This is the example of an attacker stating the position of a vehicle in the parallel street of the real location of a victim, after a minute, the victim has a range in which they could actually have been that include the forged location, and thus consider true the attacker's declaration.
Although this is not a real threat since stating position near the actual ones will provide, maybe a bit less precise but, still a valuable traffic situation.

If the position of victim is stated to be far away, not even with 99% of malicious nodes, the attack will succeed, because they are uncoordinated, and their false statements are not enough to drive the system away from the true positions.

Two are the suggested mitigations for this threat. In the simulator implemented, participants roam in a random step, probe proximity for near nodes and issue transactions at the end of the turn, this means that timestamps of transactions generated during that turn are all equals.
In a real system, however, timestamps reflect the real time of detection allowing a better estimation of distance covered; therefore favoring a prompt identification of a forged transaction before that its range could represent a viable location.
This adjustment can be easily applied modifying the formula proposed in Section 2.2.3, changing the part where it is calculated the traveled distance adopting the actual timestamp instead of turns passed.

```
2 * (Current Turn − Turn_of_Transaction) * MAXDISTANCECROSSED
```

Substituting it with:

```
(CurrentTime − Transaction.getTimestamp()) * MAXSPEED
```

Another efficient solution is to dynamically regulate the number of previous blocks that are accessed to collect information; the value can be adjusted according to the population density of the shard or the area of the map that is being monitored.
If we consider that setting `LASTBLOCKSTOCHECK` to 20 means that we are consulting also information of the past 20 minutes, then we should also keep in mind that under the same population number, greater percentages of the map are covered by participants of more populated areas, due to smaller dimensions of crowded shards.
Besides, the number of transactions will be higher in more crowded areas,

this means that is no longer required to examine remote blocks to find transactions toward a specific vehicle. Furthermore, a genuine vehicle will unlikely roam into a city without being perceived by other participants for long periods.
Reducing the time passed between the commitment and the check, transactions will have less time to "expand" their range of possible truth, hence, when false, they will be more easily identified as so.
In less populated areas, instead, the `LASTBLOCKSTOCHECK` can remain unvaried as greater distances are covered and the wider map dimensions help to mitigate the threat. The following outcomes prove this to be a sound strategy: reducing the `LASTBLOCKSTOCHECK`, false transactions are detected.

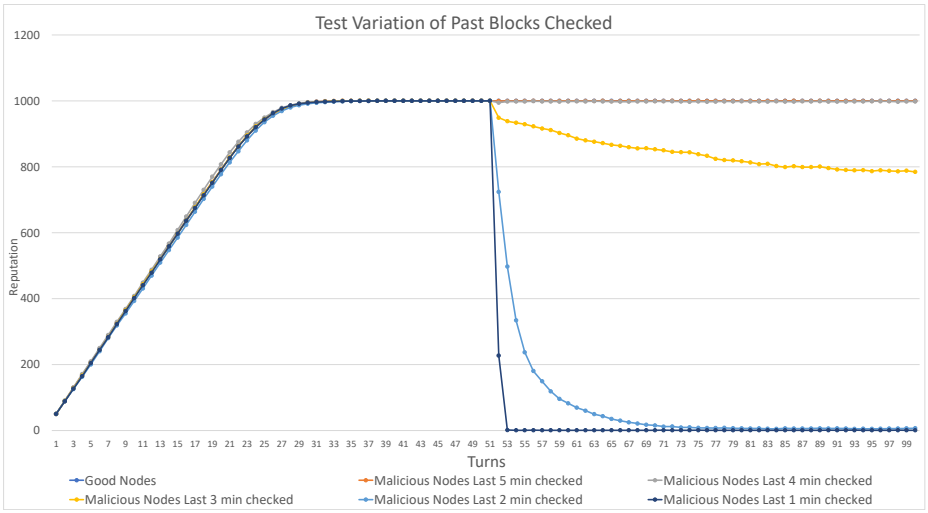


Figure 4.18: The chart shows the outcomes of the previously proposed Random Attack, this time with different numbers of older blocks checked.

The Chart 4.18 is a bit different from other tests because we tried to adapt the time flowing to apply the previous mitigation too, redimensioning parameters to cope with turns lasting one second. Transactions, indeed, are now validated, causing reputations changes, every 60 turns, but their collection is pursued every step, leading to a 60 times higher amount of transactions.
The adaptation does not modify trends but they will be scaled with a bigger factor, this is why the reputation cap is reached within a short time. When `LASTBLOCKSTOCHECK` is high, the attacks are successful, while

reducing its value, they are prevented.

We also provide the output of the same experiment but when a `BLOCKLIMIT` is applied. To make the result resemble more previous tests, the transactions limit is set to the average transactions per turn, divided by 60, so the numbers will be similar. The Chart in Figure 4.19, indeed, is more comparable to the previous ones.

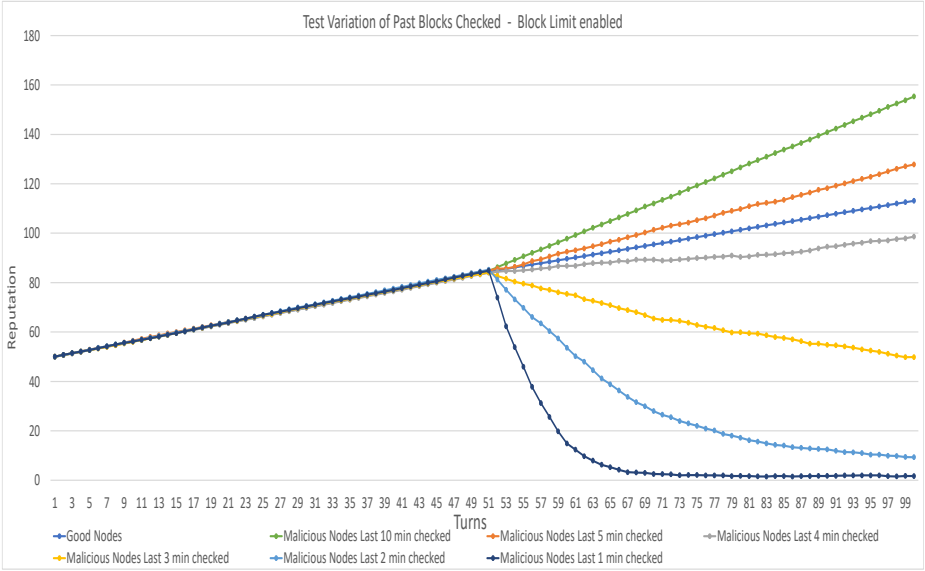


Figure 4.19: The chart shows the outcomes of the previously proposed Random Attack, the mitigation is still applied, but there is also a limit to the transactions evaluated during the validation. The chart resemble more the others, because of the similar reputation changes.

Penalty Calibration

An important parameter to be properly set is the amount of penalty. This category of tests does not actually show an attack, but rather evaluates the most appropriate number representing the penalty given to a node that has just shared an incorrect transaction.

The scenario analyzed required some inaccurate transactions to punish participants, so we applied the Random Attack for its simplicity. The input is the same as in the test case proposed in Experiment 4.16a.
Figure 4.20 shows the reputation trends of malicious nodes when different

values of penalty are adopted.
We can notice that having a penalty equal to the reward given in case of correct information sharing is not feasible since an attacker can obtain reputation from a normal behavior and keep on pursuing its attack, maintaining stable reliability.
On the contrary, an excessively harsh penalty risks to abruptly condemn an error or a malfunction of a genuine node. We deemed more important a reactive response against attackers at the possible expense of genuine nodes to be punished, so a balanced trade-off appears to be a $1 : 20$ reward-punishment ratio.

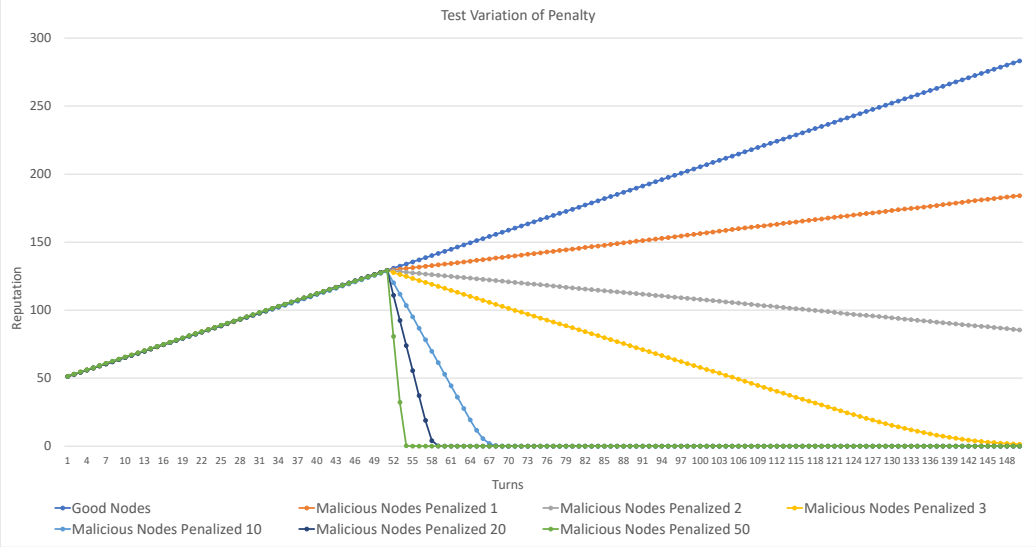


Figure 4.20: The chart shows how different values of punishment affect the average reputation of malicious nodes performing a detected attack.

4.6.2 Experiment 7: Aimed Attack

Unlike the attack proposed in test 4.6.1, this one is very powerful. It consists of distributed malicious entities that state the same fake location of the same singular target. Again, the attackers behave normally for some steps, so that they can gain reputation, and then strike altogether.

People who want to frame a particular node can lead this attack, but it must be taken into account that all participants are assumed to be pseudonymized, so there is already one layer of security on top to try and mitigate such attacks.
All these types of tests have a similar input: 1000 meters side map, 5000 participants, 150 stations, percentage of attackers equal to 10% of considered vehicles, 30 turns before strike, out of 100 turns in total. For all the attacks, the victim is always the same agent.
Four main experiments are worth to be reported in particular.

The first one is a plain attack as above descried:

```
Average transactions per turn:  10822
Number of genuine nodes:  4950
Number of malicious nodes:  550
Average reputation of all participants:  247.766724
Average reputation of genuine nodes:  220.372528
Average reputation of malicious nodes:  289.194550
Transactions of genuine nodes toward the victim:  184
Transactions of malicious nodes toward the victim:  37950
which are:  99,517494% of the total 38134 transactions
Total Reputation Changes:  1080248
Total transactions:  1082200
False positives:  102836 9,519666%
False negatives:  129 0,011942%
```

Chart in Figure 4.21

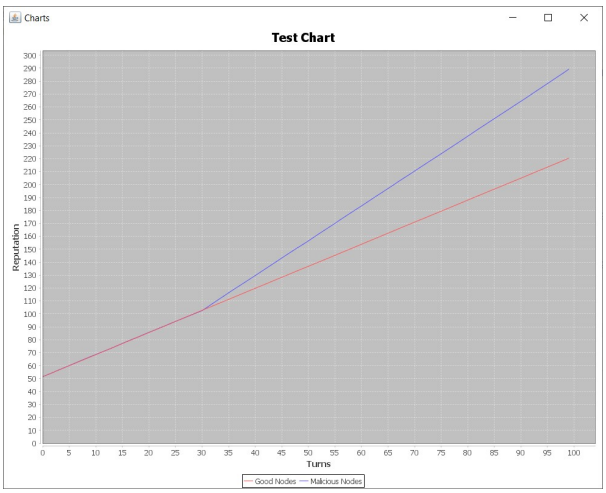


Figure 4.21: The chart shows the average reputation of normal nodes and malicious nodes, when it is performed an organized attack toward a single target. No mitigation is present. The attack succeeds.

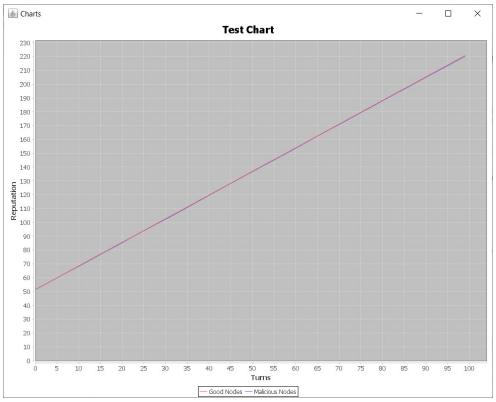
The second test introduced a first mitigation; when any node receives more than x transactions from talkers in a single turn, that information is discarded. This might seem a waste, but it is unlikely to receive those many transactions, even when x is a small percentage of the total participants. In this experiment `MAXIMUMTALKERS` was set to 250 per turn, while the average received by non-victim nodes is 5.

The most relevant differences are:

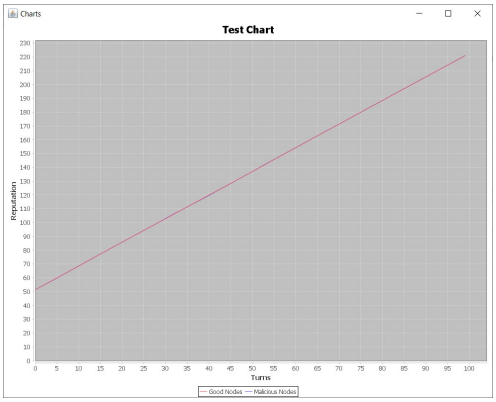
```
Average reputation of all participants:  241.354340
Average reputation of genuine nodes:  220.722626
Average reputation of malicious nodes:  220.136368
False negatives:  0 0.000000%
Total Reputation Changes:  1042070
Total transactions:  1082117
```

Chart in Figure 4.21, it can be noticed how the mitigation prevented malicious nodes to grow reputation over normal participants and, eventually, take control.

The third test added to the previous one, the draw advantage of the attackers, meaning that now, if the validator elected is a malicious node and *at least* one of the supports is malicious too, they will reward every



(a) The chart shows the average reputation of normal nodes and malicious nodes when it is performed an organized attack toward a single target. The mitigation avoid attacker to overcome normal nodes.



(b) The chart shows the average reputation of normal nodes and malicious nodes, when it is performed an organized attack toward a single target. The mitigation avoid the success of the attack, but the malicious extraction, allows attackers to have a slightly higher reputation.

Figure 4.22

node, regardless the correctness of the transactions they shared, benefiting all the attackers. Only 2 extractions were won by attackers, and the total average reputation of malicious nodes is slightly higher, under the same conditions. Figure 4.22b

The last test provides a risky solution: punish every talker of that has the target that is shared with more than `MAXIMUMTALKERS` other nodes. This solution is very effective, although some genuine nodes will be unfairly penalized when actually bumping into the real vehicle of the victim. Attackers, though, are severely punished, even in this sticky situation. Figure 4.22b

This typology is overall very successful because the probability of the target node to meet a greater number of genuine participants that confirm its real position than the attackers' number is very low.
Although this attack often succeeds, it must be considered that the target is just one node and thus its accomplishment will not deviate the traffic behavior nor the general participants' reputations.
Some mitigations have been proposed, of course, the simplest solution is to have a dense number of genuine transactions, as in every blockchain,

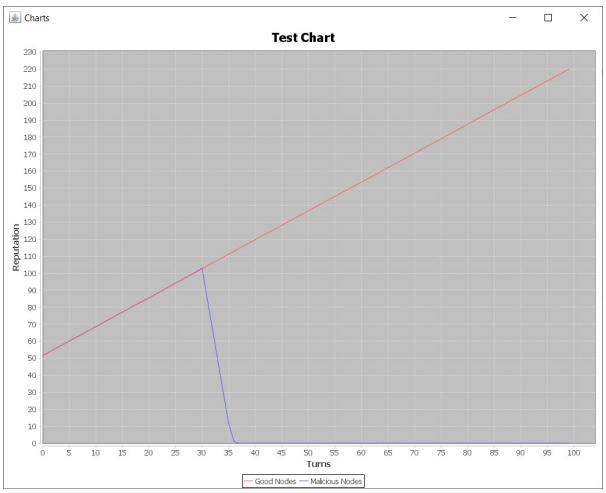


Figure 4.23: The chart shows the average reputation of normal nodes and malicious nodes when it is performed an organized attack toward a single target. The mitigation not only avoid the success of the attack but also punishes all the nodes involved in the attack.

the best defense is the highest number of good nodes as possible.
Another suggested mitigation is to limit the reputation gained after that two nodes are reporting their reciprocal position for more than a defined time. However, this idea has not revealed to be a complete solution to the problem, but just a mitigation.
After some tests, it comes up that the attack is successful only if the number of malicious transactions is above the 94% of the overall transactions having the target node as actual target, and using some stations with a high reputation as in every scenario.

4.6.3 Experiment 8: Distributed Attack

The distributed attack is the closest to a real scenario, yet is similar to the previous one. The concept is the same, but now attackers are distributing their computational power toward different chosen targets.

The consequences of a successful attack are more serious since given a large number of targets, a complete part of traffic will result in being shifted elsewhere.
However, its probability of succeeding are very low, it requires more than 97% of malicious transactions on each target, and the computational requirements are very demanding.

The input of the test is the usual of other tests: 1000 meters side map, 5000 participants, 150 stations, 15% of attackers, 30 turns before striking out of 100 turns.
In this scenario though, attackers had a computational power 5 times more powerful computational capacity (i.e. sending 5 forged transactions per turn).

Outcomes will be briefly reported without charts since they are very similar.

Table 4.3: Outcomes of Distributed attacks

Number of Targets	Target Percentage	Attackers Reputation	False positives	False negatives
2	0.036%	566.07	28.832%	0.0158%
5	0.09%	563.76	28.796%	0.0388%
10	0.18%	561.32	28.7549%	0.0818%
50	0.9%	543.95	28.7708%	0.4078%
70	1.27%	530.75	28.7094%	0.5723%
100	1.82%	0	7.5190%	0.0074%
500	9.09%	0	7.3377%	0.0000%

The attempt of applying the same mitigation of the previous test was not effective due to the distributed nature of the attack, resulting only in a slight reduction of attackers' reputation growth. Figure 4.24

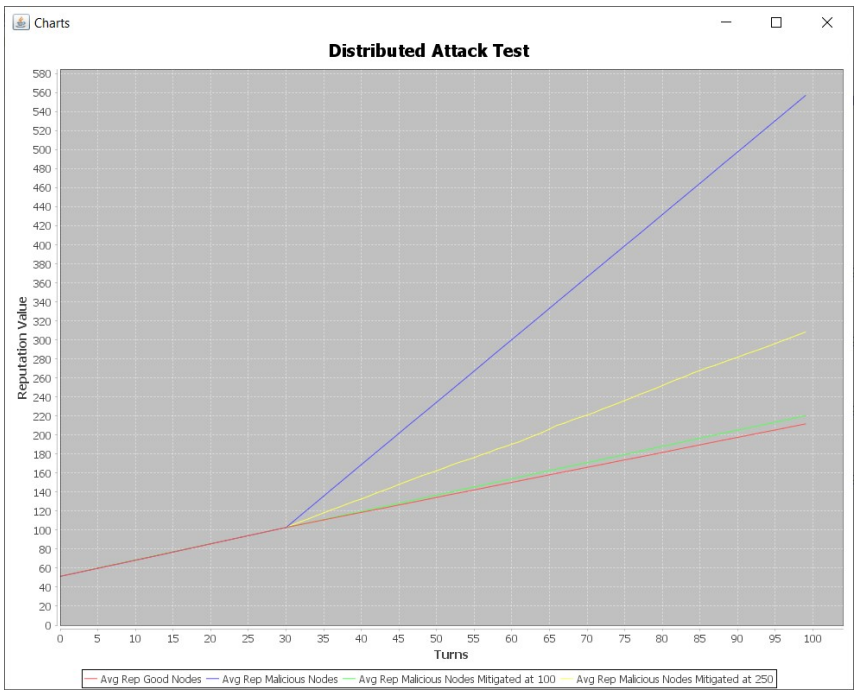


Figure 4.24: The chart shows a comparison between the average reputation of normal nodes and malicious nodes, when it is applied a mitigation with different levels of severity. The mitigation is not as effective as in previous scenarios (4.22a).

The percentage of participants deviated is very low, beyond which the attack is unsuccessful, around 1.5%.
Some test, trying to shift 10% of the participants, required more than 50% of attacking nodes each with 10 times the computational capacity of standard users.
Mitigations proposed could be the limitation of few transactions per participant each turn, so that if an attacker has enhanced computational power, it is capped and also avoid transaction spam.
Limiting the transactions per block, picking a specific number of random transactions among those sent in the current turn, did not prove to be a proper solution because, statistically, the threat is still present, but this mitigate attackers' reputation grow.
Eventually, to discourage the possibility of malicious nodes to communicate their own self-location in order to gain reputation, the control of having a target ID different from the sender's has been added.

Chapter 5

Limitations

5.1 Challenges

The research deals with topics that are quite recent, which resulted in a sort of shortage of material and previous studies. This was especially evident when analyzing specific aspects, such as little known consensus algorithms or the performance of blockchains adopting them.
The lack of previous results, data, samples, and comparisons have made sourcing more challenging. Surely, several studies have been conducted regarding smart vehicles, blockchains and consensus reaching, but not linked to each other, leaving an open questioning about the feasibility of a system that combines all together.

Moreover, many of those data were produced as a result of private and, often, commercial studies; without the disclosure of the generation procedure, they could have not been used for research purposes.
It cannot be ignored the fact that blockchain is a controversial topic, due to its application in cryptocurrencies, and so conflicting opinions, equally supported by data, had to be carefully assessed.

Similarly, although a great variety of blockchain structures and consensus algorithms have been designed, very few of them have been applied to

actual systems or frameworks. Most of all the existent cryptocurrencies, to name the main applicative scope, own very fragile credibility because of their untrusted structure.

Besides the lack of experimental data, it has been difficult to gather comparisons between other possible implementations. Some hypotheses have remained just assumptions, but that is exactly why they have been constructed as solid, restrictive and realistically as possible.

5.2 Model Limitations

The decision to implement an existing typology of blockchain helped an appropriate dimensioning of the system but could not be simulated by means of off-the-shelf software, that is why it has been necessary to implement a simulator to fulfill the framework's requirements.
Whilst the implementation from scratch allowed us to perfectly adapt simulations to the proposed algorithm, on the other hand, some shortcomings arose.

As already mentioned in chapter 4.1.3, it was necessary to adapt the decentralized framework to work on a single machine; this adaptation consists of a central database to which everyone can access. However, this structure is not as scalable as the actual protocol, because no matter how powerful is the machine simulating participants, it will not be able to replicate billions of agents taking part in the system. Especially, considering that the simulator requires a demanding loop for every node to check each other participant in order to establish who is nearby.
The tests performed were limited by this shortcoming, and it was necessary to scale the dimension and participants' number in some scenarios.
In an actual instance of the framework, though, users will be able to immediately probe the vicinity with DSRC communication.

Eventually, some limitations that come with decentralization: this feature is not only one of the greatest advantages introduced by the distributed ledger, but also, unfortunately, a decentralized implementation may expose a framework to the vulnerabilities discussed in section 1.2.4. Compared to a centralized environment, it is much harder to regulate or apply modification, and it could even lead to a hard fork with tremendous consequences for the system stability.

Chapter 6

Conclusions and Future Studies

6.1 Goals summary

The purpose of the thesis is the evaluation of a framework in its early stages. A new idea to assist traffic management has been proposed. All the elements adopted or rejected have been studied and their utility evaluated. Decentralization is the main feature we aimed to introduce and on which we calibrated all other properties. We searched for all the advantages this structure would have brought, as the focus is to provide a solid decentralized alternative in a field that has always been managed with centralized policies.
We provide an option that in addition to mitigating all the mentioned threats to which are subjected centralized systems, also features complete transparency, disintermediation, and verifiability in its properties.

We searched for vehicle components that could be useful to help communication in the system. The system required a solid evaluation technique to judge the truthfulness of the information exchanged through it, this led to the idealization of a protocol that could classify users' reliability.
The goal of the component is to assess as precisely as possible who is behaving properly and who does not, which participant is lying and who

is, indeed, contributing to the system functioning. The fact that everyone is free to lie and there is no indisputable judge whose decisions cannot be objected made this goal the most challenging one.

6.2 Research development

Previous studies and results have been analyzed, searching for possible foundations on which start to build the framework. Some material turned out to be helpful to better understand some elements involved, but nothing was published regarding a system strictly similar to the proposed one.
As regards the ledger structure, we could rely on an existing system, but the consensus algorithm, on which is based also the truthfulness assessment, had to be implemented from scratch.

To the research objectives, is, also, added the implementation of some sort of simulator to evaluate, with experiments, the feasibility of the framework and its possibility to be applied to real scenarios in terms of scalability and reactivity.
Tests have been run to check the robustness of the algorithm, but also, how resistant to inevitable threats it will face. A dimensioning was set according to the collected data on one of the most populated cities in Italy, Milan, since it would have represented a practical scenario.

6.3 Achievement Summary

The design of a consistent structure has been reached. Requirements, assumptions, and constraints have been delineated precisely so as to make the framework operating at its best; its robustness and scalability have been assessed relying on concrete and real examples. The algorithm has been successfully implemented and so has a simulator. The feasibility tests are suitable, the system is able to manage a very large number of participants, even though the possible adoption of this project will surely be gradual.
Minimal densities tested are significant and applicable to real situations:

congested cities, as well as rural areas, where not many participants are expected to roam. The initial situation has been taken into account, the best parameters calculated and some techniques designed to promote framework development.

Attack tests revealed load levels the system can handle. Thresholds found are compatible with a robust structure upon various levels of cooperated and distributed attacks; mitigations have been proposed and back-up solutions can be applied.

The system is feasible, simulations adopting its algorithm prove it.
The idea is supported by existing companies applying parts of it that count numbers of users resembling the one esteemed and used as sample in our framework.

6.4 Cope with Limitations

Overcome the limitations mentioned in sections 5.1 and 5.2 will surely require time and additional research. Topics covered in this thesis are still developing, but are getting more and more popular, attracting the interest of the scientific community. This will soon lead to more sophisticated studies, that will take advantage of proper technologies that are not so commonly available at the moment.

Speculating on the future diffusion of autonomous vehicles, equipped with appropriate computers, it will be easy to implement a framework as proposed, testing it directly on participants' devices.
Also, the spreading of autonomous vehicles, not only will promote market and companies' interest in these topics, but also will produce more research data and statistics.

6.5 Future research

The thesis itself is the feasibility study of an idea, this implies a continuation in the research. The results obtained could serve as a foundation

for a subsequent analysis or studies that cover a specific aspect shared with this framework.
In the future, a system based on the assumption and outcomes of this thesis could be effectively implemented, brought forth by a research team or an interested company willing to fund the project. The application of the framework could lead to a revolutionary management of traffic. If the project caches on, the possibilities will be countless; not just limited to traffic boarding, but also facilitating authorities and emergency vehicles, car sharing systems, self-driving cars, pedestrian safety and new technologies future will hold.

Bibliography

[1] Tesla Tap, *Processors analysis and count*. [Online]. Available: https://teslatap.com/undocumented/model-s-processors-count.

[2] Eurostat, *Statistiche del trasporto passeggeri*. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Passenger_transport_statistics/it.

[3] Google, *Google maps*. [Online]. Available: https://www.google.com/maps.

[4] Waze Mobile Ltd., *Waze*. [Online]. Available: https://www.waze.com.

[5] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf.

[6] T. Mikula and R. H. Jacobsen, "Identity and access management with blockchain in electronic healthcare records", *2018 21st Euromicro Conference on Digital System Design (DSD)*, pp. 699–706, 2018.

[7] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems", *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, Jul. 2018. DOI: 10.1109/MCE.2018.2816299.

[8] M. Bellini. (Sep. 2019). Blockchain: Cos'è, come funziona e gli ambiti applicativi in italia, [Online]. Available: www.blockchain4innovation.it/esperti/blockchain-perche-e-cosi-importante/.

[9] MLSDev, *Blockchain architecture basics: Components, structure, benefits & creation*. [Online]. Available: https://medium.com/@MLSDevCom/blockchain-architecture-basics-components-structure-benefits-creation-beace17c8e77.

[10] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy", *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, Dec. 2017. DOI: 10.1109/MCOM.2017.1700879.

[11] C. Xu, H. Liu, P. Li, and P. Wang, "A remote attestation security model based on privacy-preserving blockchain for v2x", *IEEE Access*, vol. 6, pp. 67 809–67 818, 2018. DOI: 10.1109/access.2018.2878995. [Online]. Available: https://doi.org/10.1109/access.2018.2878995.

[12] P. Romanenko. (Dec. 2018). 20 blockchain use cases for 2018 you should know, [Online]. Available: hackernoon.com/20-blockchain-use-cases-for-2018-you-should-know-f7d2919c191d.

[13] P. Fraga-Lamas and T. M. Fernandez-Carames, "A Review on Blockchain Technologies for an Advanced and Cyber-Resilient Automotive Industry", *IEEE Access*, vol. 7, pp. 17 578–17 598, 2019. DOI: 10.1109/access.2019.2895302. [Online]. Available: https://doi.org/10.1109/access.2019.2895302.

[14] G. Brambilla, M. Amoretti, and F. Zanichelli, "Using block chain for peer-to-peer proof-of-location", Jul. 2016.

[15] *Digital services, technology and consulting*. [Online]. Available: https://www.reply.com/.

[16] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better: How to make bitcoin a better currency", in *Financial Cryptography and Data Security*, Springer Berlin Heidelberg, 2012, pp. 399–414. DOI: 10.1007/978-3-642-32946-3_29. [Online]. Available: https://doi.org/10.1007/978-3-642-32946-3_29.

[17] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot", in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2017, pp. 173–178. DOI: 10.1145/3054977.3055003.

[18] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm", *2014, Annual Technical Conference, Act 14*, pp. 305–319, May 2014.

[19] M, Divya and Biradar, Nagaveni B., "Iota-next generation block chain", *International Journal of Engineering and Computer Science*, vol. 7, no. 04, pp. 23 823–23 826, Apr. 2018. [Online]. Available: http://www.ijecs.in/index.php/ijecs/article/view/4007.

[20] V. Saini. (Jun. 2018). Consensuspedia: An encyclopedia of 30+ consensus algorithms, [Online]. Available: https://hackernoon.com/consensuspedia-an-encyclopedia-of-29-consensus-algorithms-e9c4b4b7d08f.

[21] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014.

[22] NEM Foundation, "Nem, technical reference", pp. 26–41, Feb. 2018.

[23] F. Gai, B. Wang, W. Deng, and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network", in. May 2018, pp. 666–681, ISBN: 978-3-319-91457-2. DOI: 10.1007/978-3-319-91458-9_41.

[24] V. Arasev, "Proof-of-authority network", Sep. 2018. [Online]. Available: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper.

[25] Internet of Service Foundation, "Internet of services: The next-generation, secure, highly scalable ecosystem for online services", *International Journal of Engineering and Computer Science*, Dec. 2017. [Online]. Available: https://github.com/iost-official/Documents/blob/master/Technical_White_Paper/EN/Tech_white_paper_EN.md.

[26] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle", *Black Hat USA 2015*, Aug. 2015.

[27] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, Dec. 2018. DOI: 10.1109/tits.2018.2797529. [Online]. Available: https://doi.org/10.1109/tits.2018.2797529.

[28] Politecnico di Milano, "On the suitability of blockchain for use in vehicular settings", *Politecnico di Milano*, 2018.

[29] *Bitcoin, litecoin, namecoin, dogecoin, peercoin, ethereum stats.* [Online]. Available: https://bitinfocharts.com/.

[30] K. Li. (Jan. 2019). The blockchain scalability problem & the race for visa-like transaction speed, [Online]. Available: https://hackernoon.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44.

[31] *The statistics portal for market data, market research and market studies.* [Online]. Available: www.statista.com.

[32] J. Ibanez-Guzman, C. Laugier, J.-D. Yoder, and S. Thrun, "Autonomous driving: Context and state-of-the-art", in. Mar. 2012, pp. 1271–1310, ISBN: 978-0-85729-084-7. DOI: 10.1007/978-0-85729-085-4_50.

[33] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states", *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011. DOI: 10.1109/JPROC.2011.2132790.

[34] NEO Fundation, "Neo white paper", Jul. 2017. [Online]. Available: https://github.com/neo-project/docs/blob/master/docs/en-us/basic/whitepaper.md.

[35] P. Bartolomeu, E. Vieira, and J. Ferreira, "Iota feasibility and perspectives for enabling vehicular applications", Aug. 2018. DOI: 10.1109/glocomw.2018.8644201. [Online]. Available: https://doi.org/10.1109/glocomw.2018.8644201.

[36] *The crypto exchange you can count on.* [Online]. Available: https://www.blockchain.com/.

[37] *Comune di milano, mobilità area urbana di milano*, https://www.comune.milano.it/aree-tematiche/mobilita, Accessed: 2019-09-30.

[38] *Atlas of urban expansion - areas and densities of milan.* [Online]. Available: http://www.atlasofurbanexpansion.org/cities/view/Milan.

[39] M. Giorgio Goggi Ivan Genovese, "Studio sulla congestione del traffico a milano e in altre citta' comparabili e valutazione dell'efficacia dei rimedi", *Mobility Conference 2012*, Jan. 2012. [Online]. Available: http://mobilityconference.it.

[40] Z. Peng and Y. Chen, "All roads lead to rome: Many ways to double spend your cryptocurrency", *CoRR*, vol. abs/1811.06751, Nov. 2018. arXiv: 1811.06751. [Online]. Available: http://arxiv.org/abs/1811.06751.