An On-Chain Governance Model Based on Particle Swarm Optimization for Reducing Blockchain Forks

by

Reza NOURMOHAMMADI

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, DECEMBER 7, 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Kaiwen Zhang, Thesis supervisor
Department of Software and Information Technology Engineering, École de technologie supérieure

M. Bora Ung, Chair, Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

M. Chamseddine Talhi, Member of the Jury
Department of Software and Information Technology Engineering, École de technologie supérieure

M. Mohammad Hamdaqa, External Examiner
Department of Computer Engineering and Software Engineering, Polytechnique Montréal

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON NOVEMBER 29, 2022

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

First and foremost, I would like to thanks to my supervisor, Prof. Kaiwen Zhang, for his guidance throughout my PhD program. I have always been supported by him through thick and thin. Through his knowledge and experience, as well as his attitude toward high quality research, he has guided me in the right direction over the past few years. It is an honor to have him as my advisor, and I feel very fortunate to have him. My sincere gratitude goes out to you for your enormous support and patience during my doctoral studies.

Then I would like to thank all the people who contributed in some way to the work described in this thesis.

I would like to thank my friends and colleagues who supported me during my time here.

I am also very grateful to all those at the École de technologie supérieure university, who were always so helpful and provided me with their assistance throughout my dissertation.

Lastly, I also thank my family who encouraged me and prayed for me throughout the time of my research. My parents have always encouraged me to pursue my goals. Everything from my name to my life was given to me by them. When it comes to describing how much I love and appreciate them and the efforts they have made toward making my life what it is today, they truly shut me up. It is because of them that I did this; it is because of them that I strive to do better. It is my primary goal in life to make them proud of me. It has been my goal to make them think, "I am proud of my son," as I have stated numerous times.

Une Blockchain auto-configurable adaptative basée sur des techniques d'apprentissage automatique

Reza NOURMOHAMMADI

RÉSUMÉ

La Blockchain est une technologie émergente qui a rencontré un certain nombre de défis, y compris des fourches et de faibles taux de traitement des transactions par rapport à d'autres systèmes de paiement. La plupart des systèmes publics de blockchain sont enclins aux fourches comme un processus naturel. À la suite de la résolution de ces problèmes, les transactions latérales sont rejetées lors de la réorganisation de la chaîne (réaménagement). Le retard est causé par l'obligation pour les clients d'attendre plus de confirmations avant de terminer les transactions. Pour améliorer la sécurité, la vitesse et l'efficacité d'un réseau de blockchain, il est souhaitable de réduire la probabilité de forking autant que possible. Il est nécessaire de comprendre le comportement des systèmes de blockchain pour optimiser les paramètres en conséquence. Nous proposons ici un nouveau modèle de fourche qui intègre des paramètres qui n'ont pas été pris en compte précédemment, tels que le retard dans le réseau et le degré de validation. Pour vérifier la validité de notre méthode proposée, nous avons mené une série d'expériences utilisant le simulateur BlockSim sur le réseau Ethereum et la spécification EIP-1559. Une fourche est moins susceptible de se produire si le degré de validation est diminué et le coût marginal des mineurs est augmenté (comme dans EIP-1559). D'après les résultats de cette partie, la probabilité qu'une fourche se produise est réduite d'environ 10%. De plus, les résultats expérimentaux démontrent que notre méthode est très précise dans la prédiction de la probabilité de la fourchette.

Sharding a été développé pour résoudre le problème des faibles taux de traitement, mais il améliore également l'évolutivité du réseau. Toutefois, on ne sait toujours pas comment cela peut affecter la probabilité d'une fourchette. Deuxièmement, la thèse tente de déterminer comment l'ajout de nouveaux shards à une blockchain affecte la probabilité de fourches. Comme première étape vers la réalisation de cet objectif, nous avons développé un nouveau simulateur qui facilite la simulation de réseaux cloisonnés. Par la suite, nous avons examiné les effets du requin sur la présence de fourches. Les expériences multiples ont été conduites sur les réseaux permis de deux EIP-1559 avec 60 et 120 nœuds, respectivement. Selon notre étude, l'ajout d'un shard entraîne une réduction des blocs orphelins de 60% en moyenne. De plus, nous proposons un modèle de probabilité de fourche qui réduit les fourches de 23% et 15%, respectivement, pour les réseaux à 60 et 120 noeuds.

Comme l'un des inconvénients les plus significatifs associés aux réseaux de blockchain, la vitesse de traitement est un inconvénient important. Cela peut se faire par le biais de l'affûtage, qui a la capacité de résoudre ce problème. Il en résulte que l'évolutivité du réseau peut être améliorée. Il était difficile de déterminer comment le requin aurait une incidence sur la probabilité que des fourchettes surviennent à la suite du requin dans cette étude. Pour atteindre cet objectif, nous avons mené un certain nombre d'expériences sur le réseau EIP-1559 en utilisant 120 noeuds dans la troisième partie de cette thèse. Selon notre analyse, l'ajout d'un shard au système entraîne une diminution de 60% du nombre de blocs orphelins. Un nouveau modèle de gouvernance en chaîne a également été mis au point, qui utilise l'optimisation des essaims de particules pour réduire la probabilité de fourches entre les différents gisements. À la suite de notre étude, nous sommes convaincus que le modèle de gouvernance en chaîne proposé réduit les risques associés à la fourchette et maintient des expériences positives pour les utilisateurs.

La thèse contribue en fournissant une blockchain adaptative avec des capacités d'apprentissage pour les blockchains publics. Cela réduira la probabilité que des fourches se produisent et, par conséquent, se traduira par une architecture de blockchain évolutive. Cette solution permet aux réseaux d'apprendre la configuration optimale en fonction des entrées de données locales qu'ils reçoivent. Selon les résultats expérimentaux, la solution proposée a obtenu de meilleures performances et une meilleure évolutivité que la solution de pointe.

Mots-clés: Les Chaînes de blocs, Ethereum, Sharding, Degré de validation, Gouvernance en chaîne, Fork, Optimisation des essaims de particules, EIP-1559

An On-Chain Governance Model Based on Particle Swarm Optimization for Reducing Blockchain Forks

Reza NOURMOHAMMADI

ABSTRACT

Blockchain technology has emerged as one of the most promising technologies of the past few years. However, it has encountered a number of challenges, including forks and low processing rates when compared to other payment systems. It is a natural process for most public blockchain systems to fork over time since forks are a natural part of the process. In order to resolve these issues, the side fork transactions will be discarded during chain reorganization (reorg) after these issues have been resolved. As a result of the requirement for clients to wait for more confirmations before being able to complete transactions, there is a slight delay in the process. The probability of a blockchain network forking to the greatest extent possible should be reduced as much as possible. This will improve the security, speed, and efficiency of the blockchain network. To the greatest extent possible. Understanding the behavior of blockchain systems is crucial to optimizing parameters. In this reserach, we propose a novel fork model that incorporates previously unconsidered parameters, such as the network delay and degree of validation. We have conducted a series of experiments using a blockchain simulator on the Ethereum network and the EIP-1559 specification to verify the validity of our proposed method. It is less likely that a fork will occur if the validation degree is decreased and the marginal cost of the miners is increased (as in EIP-1559). This part indicates that the probability of a fork occurring is reduced by approximately 10%. Further, the experimental results demonstrate that our method is highly accurate in predicting forking probabilities.

By developing sharding, we were able to address the problem of low processing rates, but it also enhances the scalability of the network by making it more efficient. There are still some questions that need to be answered, however, regarding the impact of this on the likelihood of a fork. The second objective of this thesis is to determine whether adding new shards to a blockchain impacts the likelihood of forks in that blockchain as a whole. As a first step towards achieving this goal, a new simulator that facilitates simulation of sharded networks has been developed as a first step towards achieving it. We then examined the impact of sharding on the occurrence of forks as a result of sharding. There have been a number of experiments conducted on two EIP-1559 enabled networks, each with 60 and 120 nodes, in which a number of experiments were carried out. As a result of adding one shard, it has been found that 60% of the orphan blocks are reduced on average when one shard is added. Furthermore, we present a model that reduces forks by 23% and 15%, respectively, when there are 60 or 120 nodes in the network.

The processing speed of blockchain networks is one of the most significant disadvantages associated with these networks. In order to achieve this, it may be necessary to implement sharding, which is capable of solving this problem. This will lead to an improvement in the scalability of the network as a result. It was difficult to determine how the sharding process might impact the likelihood of forks arising as a consequence of the sharding process in the current study.

In the third part of this thesis, we performed a number of experiments using 120 nodes on the network EIP-1559 to achieve this objective. Our analysis indicates that adding a shard to the system results in a 60% reduction in the number of orphan blocks. There has also been the development of a novel on-chain governance model that uses particle swarm optimization in order to reduce the likelihood of forks between different shards. Based on the results of our study, we are confident that the proposed on-chain governance model reduces the risks associated with forking and maintains a positive user experience.

As a contribution to public blockchains, the thesis provides an adaptive blockchain with learning capabilities. Consequently, there will be a reduction in the likelihood of forks occurring, leading to a more scalable blockchain architecture. Based on the local data inputs that networks receive, this solution enables networks to learn the optimal configuration. Based on the experimental results, the proposed solution has demonstrated better performance and scalability than the current state of the art.

Keywords: Blockchain, Ethereum, Sharding, Validation degree, On-chain Governance, Fork, Particle Swarm Optimization, EIP-1559

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

EIP	Ethereum Improvment Proposal
PSO	Particle Swarm Optimization
MOPSO	Multi Objective Particle Swarm Optimization
DOS	Denial Of Service
POS	Proof Of Search
POW	Proof Of work
DPOS	Delegated Proof Of Stake
POAI	Proof of Artificial Intelligence
ATN	Average Transaction Number
NN	Neural Network
POS	Proof Of Stake
AI	Artificial Intelligence
PBFT	Practical Byzantine Fault Tolerance
VRF	Verifiable Random Function
DAO	Decentralized Autonomous Organization

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

D	Network delay
t_v	Verification time
t_inv	The arrival time of the *inv* message in destination node
t_get_header	A message to the origin node asking for the newly mined block header
t_header	A message sent by the origin to the neighbor with the new block header
$t\_get\_block\_body$	Message when the receiver node asks the sender to send block body
t_block_body	Arrival time of the message *block_body* sent by the origin node
rho	Security assessment rate
tau	Synchronization time
rs	Stale block rate
t_prop	The time required for a block to propagate over the network
t_B	The time required to mine a new block
$rs1$	Fork evaluation model
$v(t+1)$	The velocity of the particle at time *t*
$v(t)$	The velocity of the particle at time *t+1*
$c1$	Learning factors to control the exploring capability of PSO
$c2$	Learning factors to control the exploiting capability of PSO
w	The inertia weight coefficient
$x(t)$	The position of the particle at time *t*

$x(t+1)$	The position of the particle at time *t+1*
p_gbest	Position of the best particle that has the best fitness value in each iteration
p_best	The best position of the particle from the beginning to current iteration
$block\_interval$	Time interval between blocks
$rs3$	Fork evaluation model
V	Validation degree

INTRODUCTION

As a result of the Bitcoin cryptocurrency, the first blockchain application was introduced, which was then applied to other applications (Bravo-Marquez, Reeves & Ugarte, 2019). Blockchain technology implements decentralized ledger technology due to the lack of trust between the nodes to transfer information and data. Blockchain is an existing implementation of decentralized ledger technology. Using a distributed ledger, parties can share data and information in a trustless environment. As the name implies, a distributed ledger is a database that is located across multiple locations or is shared by a number of participants. (Nofer, Gomber, Hinz & Schiereck, 2017; Gupta, 2017).

There are also some unique characteristics of a blockchain, such as immutability and security. Blockchain transactions cannot be changed or altered once they have been recorded because each participant maintains a copy of the ledger (Li, Jiang, Chen, Luo & Wen, 2020b). It is also possible to tokenize using blockchain regarding privacy-preserving issues, which can prove to be advantageous for a wide range of applications. Blockchain technology has the main potential benefit of disintermediating an untrusted network, enabling participants to participate in a reliable peer-to-peer network without the need for third parties.

Fork problems

A fork occurs when two or more miners propagate their blocks simultaneously. Thus, each node accepts the earlier block as the chain tip, resulting in the growth of two or more chains at once. The decentralized nature of blockchain causes some inconsistencies in the network. There are several negative aspects to these inconsistencies; they reduce the trustability and reliability of the network as they lead to differences in local copies of the Blockchain and make the network more prone to forking.

The majority of applications use blockchain technology, so analytical modeling and simulation are essential for evaluating performance and observing behavior. There have been relatively few efforts to simulate blockchains. There are very few publications in the literature, and almost all of them concern only the analytical modeling of Bitcoins (Memon, Li & Ahmed, 2019).

Blockchain systems are being developed for a variety of applications, but there are no tools available for evaluating them. The most common method of analyzing the performance of a system is through emulation, which simulates its behavior across a large set of machines (Dinh *et al.*, 2017).

Despite the significant overhead associated with this approach, it is not scalable for use in real-world deployments. Furthermore, it is necessary to consider the power consumption of large-scale systems. Simulators can be used as a solution. Simulators for network and distributed systems play an essential role in the evaluation of protocol and system performance under a variety of conditions. Protocols can be implemented and deployed more easily with the aid of simulators. Simulations enable the study of large-scale systems containing thousands of nodes in a single machine and the gathering of results within a reasonable timeframecitep (Faria & Correia, 2019).

It is important to note that forks do not turn into alternative competing realities in a healthy blockchain. In the case of a canonical chain, the truth is recognized by the entire community. It is common for others to be pruned and forgotten. It is the responsibility of the nodes who fail to acknowledge the canonical chain to relinquish the community. There are different types of finality in the current blockchain ecosystem.

Probabilistic Finality Blockchains based on proof of work, such as Bitcoin and Ethereum, cannot provide this guarantee. It is always possible that a group of nodes will gather sufficient computing power to produce a longer chain that will replace the canonical chain in its entirety. Due to the increasing difficulty of creating new blocks than the canonical one, as time passes, this is similar to a hostile takeover. However, due to the fact that the attacker would have to create more blocks than the canonical one, the opportunity to do so diminishes exponentially. It can be concluded that finality is probabilistic in this case: assuming a certain number of blocks have passed since a block was recorded, we can be certain that the block will remain permanent, and its transactions will be final.

Economic Finality The notion of economic finality is also used in POS chains. Blocks are proposed by creators and voted on by nodes. Using this model, an attacker must gather enough votes to prevent the inclusion of a valid block (incorrect vote) or authorize the creation of a parallel chain with different blocks at the same height (equivocation). In the event that someone finds these discrepancies, they can challenge the malicious voters and take their stake. This penalty is designed so that an attack is economically so expensive that it is unlikely to succeed.

Absolute Finality In Substrate-based chains such as Polkadot, Kusama, there is this kind of finality. As a result of a technical decision in the chain's protocols, after a given point, the chain will be immutable. The number of the last finalized block is determined periodically by a dedicated distributed protocol. A block after this level can be reorganized and have different transactions in an alternative chain, but those before are immutable.

Scalability

There are times when Ethereum miners appear to coordinate their actions in order to resolve hard forks or increase the maximum block size. In this case, the goal is likely to be to maximize the health of the Ethereum network. Transaction fees may have decreased as a result of an increase in computations and communication associated with the processing of blocks. However, this may have been offset by an overall reduction in centralization risk. Whether miners will use this coordination capability to pursue their own interests remains to be seen. This is in contrast to those that are in the interests of the network as a whole. There is a greater risk associated with unpredictable strategies (Roughgarden, 2020).

The issue of transaction fees dominating block rewards has been well documented, for example the incentive for a miner to launch an undercutting attack by forking a block with an unusually large number of transaction fees (Carlsten, Kalodner, Weinberg & Narayanan, 2016). Also EIP-1559 reduces the importance of transaction fees to miners by redirecting them to the network, reducing the attractiveness of such attacks (Roughgarden, 2020). It is possible to address all of these possibilities through the design of an adaptive configuration.

Ethereum's previous fee system poses a difficult design challenge. Fees are set through the use of a simple first-price auction mechanism, which is the primary point of contention. The miners select the highest priced entries for inclusion in the blockchain based on the block capacity constraints. This is in accordance with how much each user is willing to pay for their transactions to appear on the blockchain. Because first price auctions are non-truthful, selecting an appropriate bidding fee can be a challenging task and users may end up significantly overpaying to participate in the system (Leonardos, Monnot, Reijsbergen, Skoulakis & Piliouras, 2021).

The issue has recently been addressed by a new proposal (EIP-1559) (Buterin *et al.*, 2019). As part of this mechanism, a base fee is introduced, which is automatically adjusted by the protocol depending on the level of network congestion. As a result of this base fee, supply and demand are effectively matched through a reserve price. Importantly, this base fee is burned, preventing perverse incentives where miners could extract greater fees from users by engaging in dishonest behavior. To ensure that their transactions are included as quickly as possible, users may supplement the base fee with a tip, which is the only fee that the miners receive (Leonardos *et al.*, 2021). The EIP-1559 has been evaluated in terms of its economic properties, e.g., it provides incentives to both myopic miners as well as to users citeproughgarden2020transaction. The economic analysis alone is not sufficient to provide insight into whether the conditions will be met in practice, since one must examine the evolution of the parameters of the mechanism over time in order to provide insight into whether the conditions will be met (Leonardos *et al.*, 2021).

Sharding

In contrast, Bitcoin can confirm only seven transactions per second, whereas Visa can confirm 24,000 transactions per second (Georgiadis, 2019). The low throughput of blockchain systems is due to the fact that every miner validates all transactions simultaneously, so every transaction is validated simultaneously by each miner. In exchange for transaction fees, miners may receive rewards. As soon as a block is confirmed, the creator has the right to receive the fees associated with the transactions within it. Because of this, all miners are willing to pay higher fees in order to validate transactions. Assume that several transactions arrive at the same time and are sorted according to the transaction fees that each miner has paid. There is a generation of blocks that confirm the same set of transactions (Tao *et al.*, 2020).

It has been proposed that sharding may be used as a solution to this problem. By dividing the network into multiple smaller groups, multiple sets of transactions can be validated and confirmed at the same time (Kokoris-Kogias *et al.*, 2018; Zamani, Movahedi & Raykova, 2018).

It is the responsibility of several miners to validate transactions from different shards simultaneously. While this is the case, cross-shard communication is necessary in order to validate transactions since transactions from multiple shards may be required to validate a transaction. In the case of transactions conducted by users A and B in different shards, the miners in these shards will only have partial records of the transactions. The miners must exchange individual validation results between themselves in order to validate the transaction jointly, which differs from non-sharding systems (Tao *et al.*, 2020).

Blockchain governance

Despite the widespread use of blockchain technology, concerns remain regarding whether decisions regarding the technology are made in an efficient and trustworthy manner. Several infamous incidents have occurred on Ethereum and Bitcoin, two of the most renowned blockchain platforms. It was discovered in 2016 that Ethereum's smart contracts used for operating a DAO (Decentralised Autonomous Organization) project had been exploited by malicious attackers, resulting in significant economic losses. A hard fork was implemented by Ethereum in response to the DAO attack, which enabled the recovery of over 60 million USD in stolen tokens (Atzei, Bartoletti & Cimoli, 2017).

In addition, Bitcoin split following a lengthy debate over the block size (from August 2015 to January 2016) (De Filippi & Loveluck, 2016). To orchestrate a clear decision-making process in a decentralized system involving multiple stakeholders, governance mechanisms are essential. As a result of software bugs being fixed without causing a hard fork, blockchain platforms need to be updated, or stakeholders need to coordinate efficiently and effectively in order to reach a consensus that maintains the principle of decentralised decision-making (Liu, Lu, Yu, Paik & Zhu, 2022).

A blockchain governance structure and process ensures compliance with legal regulations and ethical responsibilities during the development and use of blockchain technology (Liu, Lu, Zhu, Paik & Staples, 2021b). The importance of this topic for enhancing the trustworthiness and efficiency of blockchain technology has made it a hot topic of discussion. In spite of this, existing IT/data governance frameworks and standards cannot be applied to blockchain technology, as blockchain does not declare a central authority (Liu *et al.*, 2022).

Several recent publications have examined governance structures for blockchain platforms, focusing either on custom governance methods for permissioned blockchains/citep[bao2019auditable, beck2018governance] or on financial regulatory issues (Nabilou, 2020). The current blockchain governance frameworks, however, provide limited guidance to stakeholders and the broader community that is interested in this topic due to the fact that they emphasize scattered governance mechanisms without a clear link between stakeholders and processes (Liu *et al.*, 2022).

This study examines the factors and parameters influencing the likelihood of fork occurrences, followed by a modeling attempt that incorporates all these factors.

In order to achieve this, a simulator was presented in order to simulate the blockchain. Additionally, it was enhanced with the capability to simulate the last Ethereum update regarding network fees, known as EIP-1559. As a result, we can evaluate its effects on the Ethereum network and the various methods available. The scalability of this network as well as its impact on the likelihood of a fork should be carefully examined. In addition, the model will make it easier to validate the sharding method, which is considered to be an appropriate scalability method.

Finally, based on the results obtained from the previous steps, a novel and innovative blockchain network design has been developed. By using the PSO algorithm, which has been developed to learn and modify the basic parameters in response to forks, this design can be evaluated and dynamically updated. This model can be used to reduce the likelihood of forks occurring. It is possible to create the most optimal configuration for the blockchain network in different modes due to the dynamic and flexible model provided for the blockchain network.

Research Motivation

In this section, a few examples are introduced to motivate our work in practice. By using the proposed model, it is possible to observe the behavior of an entire blockchain system, both as a mining pool and as a memory.

In this study, we demonstrate that a range of performance indices can be calculated to optimize existing cryptocurrencies, forecast mining capacity, and calculate rewards for mining power.

It is also possible to compare various types of cryptocurrencies. This model can be used to estimate power utilization and required mining power.

The behavior of a system can be determined by simulating a variety of applications. A number of factors are considered in this regard, such as information retrieval latency, waiting times, queue times, processing times, throughput, power consumption, and response times.

Due to the fact that sharding validates transactions in parallel, blockchain systems are able to perform better. Despite this, if the blockchain is to be randomly sharded, it must be communicated between shards frequently.

Problem statement

Fork Challenges: The use of blockchain, as well as other technologies, is associated with some serious challenges. A fork occurs when two (or more) miners propagate their blocks simultaneously on a blockchain network. As a result, every node will accept the block that arrived earlier as its chain tip. Consequently, two (or more) branches of the chain will grow simultaneously. For the purpose of resolving this conflict, the longest chain will be selected and the remaining blocks will be discarded as orphans. It is possible that security issues will arise in the event of a fork in the network.

It is possible that these issues may cause users to lose trust in the network, which would result in a decrease in its value. It is the network delay that plays the most significant role in causing a fork. Due to network delays and traffic, newly created blocks may reach the network nodes earlier or later. In this regard, which of the blocks arrives first determines which node receives the last block. Generally, there are two types of forks: intentional forks and accidental forks. By forking intentionally, technical problems are resolved, resources are recovered, and new features are added (Chang, Park, Wuthier & Chen, 2019).

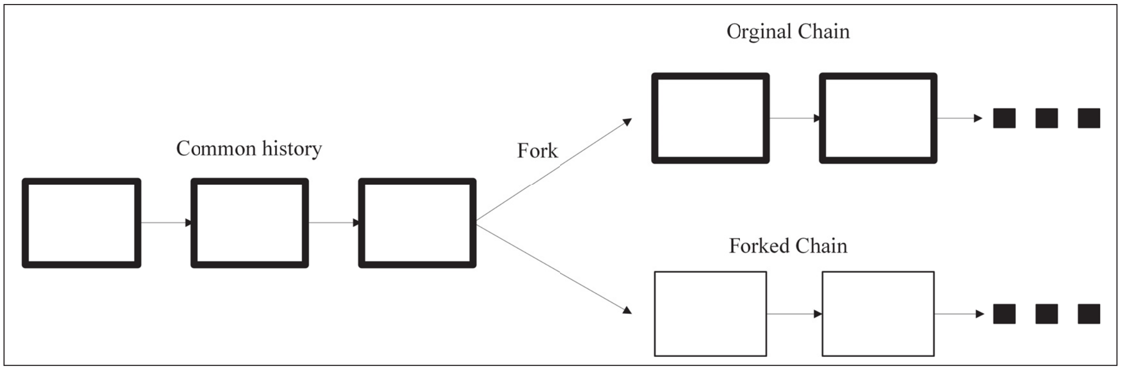


Figure 0.1 The forking of a blockchain's main chain

The Figure 0.1 illustrates the concept of a fork. The most common cause of accidental forks in a blockchain system is network delays. There may be conflicting block mining events that result in the growth of two or more branches at the same time. Branching occurs when there is a break in the main chain. In such a case, the network resolves the conflict by selecting the longest chain and discarding any orphan blocks from other chains that are not part of the longest chain. Blockchain transactions are permanently recorded after a certain number of blocks (called confirmations) are entered by the client. Ethereum has forked seven blocks in the past, and customers usually have to wait between 12 and 14 confirmations before they can use the cryptocurrency.(Narayanan, Bonneau, Felten, Miller & Goldfeder, 2016). As part of this study, the critical parameters that influence the probability of forking will be modeled and analyzed. As of now, we will refer to an accidental fork as an accidental fork.

To reduce the fork probability in a blockchain by providing a fork-free and adaptive design, the protocol should be adjusted as a result of receiving input from users and the network. Also, this enables the protocol to obtain an exact estimation of actual parameters based upon the rapidly changing requirements (Salimitari, Joneidi & Chatterjee, 2019).

Scalability: The slow rate at which transactions are processed is another challenge faced by blockchain systems. Approximately 15 transactions per second can be processed by Ethereum's network. There are other payment systems that can process thousands of transactions per second, such as VISA (Georgiadis, 2019; Gao, Kawai & Nobuhara, 2019). With an increasing number of nodes, the problem becomes more complex. There is also the issue of scalability when it comes to blockchain networks. To solve this problem, Elastico has introduced a concept called sharding through (Luu *et al.*, 2016). Sharding involves dividing nodes into groups that operate simultaneously, so that each group has its own chain of nodes. The sharding of a database can lead to an increase in transaction processing capacity. There are generally two types of transactions that occur in a sharded network:

- Intra-shard transactions are transactions between nodes within a single shard
- A cross-shard transaction refers to a transaction between nodes in different shards

A sharded network confirms transactions in the same manner as a non-sharded network. In the case of cross-shard transactions, however, the transaction must be confirmed on both shards in order to be included in the block. Therefore, confirmation of these types of transactions takes a longer period of time. We are investigating the effect of sharding on the probability of forks in this study. In order to achieve this, we have enhanced Blocksim (Faria & Correia, 2019) by adding the capability to simulate sharded networks.

On-chain Governance: Governance is a mechanism used to control and govern a network in order to ensure its stability and security. As of this point, several governance models have been introduced, which are divided into two main categories: on-chain governance and off-chain governance. Using an off-chain governance model hinders decentralization in a blockchain, because decisions are made by developers outside of the chain (Cao *et al.*, 2021). The network may undergo a hard fork if there is disagreement about off-chain governance. As an example, a disagreement over the size of blocks within the Bitcoin network in 2017 led to bitcoin cash and bitcoin being split off from one another.

By contrast, decisions are made by users through a voting process under the on-chain governance model. Consequently, these types of governance models provide a greater level of security and transparency (Miyachi & Mackey, 2021). A novel on-chain governance model based on Particle Swarm Optimization algorithm is presented in this study in order to maintain the network's stability in terms of fork occurrence probability and user experience.

Contributions

Briefly, the thesis proposes an adaptive blockchain with learning capabilities to improve the overall performance of public blockchains and reduce fork probabilities. The following contributions are made in order to address the significant and challenging issues outlined above:

1. Simulations of new and significant parameters, such as *validation degree* and EIP-1559 networks are being undertaken to determine the effect on the fork issue. The published paper will discuss and address this contribution in detail in chapter 3.
2. Introducing a new simulator that can simulate EIP-1559's multiple shard network. The new simulator, reviewed in chapter 4, introduces these parameters and describes in detail how they affect forking probability.
3. Analyzing the effects of sharding on fork probability. Furthermore, in chapter 4, it is discussed how sharding may influence the likelihood of a fork.
4. Developing a model to optimize the optimization algorithm based on fork probability and user experience. A multiobjective optimization model is presented in detail in chapter 5 in order to find a balance between the user experience and the likelihood of forks occurring.
5. Presenting a novel method of governance on the blockchain based on particle swarm optimization. Furthermore, chapter 5 presents an on-chain governance protocol for blockchain networks that is based on the PSO algorithm.

Thesis Organization

The remainder of this document is organized as follows. Our first step is to provide a brief description of the related work for each contribution in Chapter 1. Secondly, we provide in Chapter 2 a brief overview of a variety of background knowledge, concepts, and techniques necessary for a comprehensive understanding of all aspects of this thesis. The core sections then focus on one of the contributions: Chapter 3 discusses fork probability modeling of blockchains and how EIP-1559 impacts this, Chapter 4 discusses the sharding and scalability features and the impact they have on the fork rate, and Chapter 5 discusses a new governance model on chain based on PSO for reducing forks. For each section, we provide background information necessary to understand the particular subject matter. Additionally, each chapter contains experimental evaluations. We conclude our study with a summary of our findings and an outlook on future work.

CHAPTER 1

RELATED WORK

The purpose of this section is to review papers that fall into these four categories. We will begin by reviewing blockchain simulator papers and other related papers related to blockchain network simulations and blockchain modeling. Next, we will discuss articles that provide an analysis of the behavior of different blockchains and their evaluation and modeling from various perspectives.

Research on blockchain simulators has been limited. The majority of cases focused on blockchain analysis, smart contracts, and security issues. It is generally recognized that research in this field can be divided into two categories: blockchain simulation and blockchain behavior analysis. The majority of previous research has not examined the effect of sharding or on-chain governance on the probability of forks occurring, which is one of the most critical issues facing blockchain technology. In order to overcome this disadvantage, we propose to evaluate the likelihood of forks occurring in sharded networks utilizing a novel on-chain governance model. Therefore, in the next four subsections, we provide a brief overview of the corresponding papers published in recent years.

The third section will provide an overview of sharding, as well as papers that have been published in this field. In addition, they will discuss the methods that have been proposed for ensuring scalability on different blockchains, especially on the Ethereum blockchain. A variety of papers have been published in this area regarding the evaluation of security, scalability, performance, and even different sharding methods, which we will review in this section.

The final part of this section will address issues related to regulation and governance in different blockchain systems. Also, we will discuss various papers that have discussed various methods for consensus and governance in blockchain networks in this section.

1.1 Blockchain simulators

In (Aoki *et al.*, 2019), an event-driven blockchain network simulator is presented, in which each neighbor node selects the peer-to-peer overlay based on the selections of its neighbors. Despite the fact that the program cannot simulate mining in detail, it is capable of simulating the generation of blocks based on the computing capacities of the nodes. Table 1.1 shows the parameters that was used in the designing of the simulator.

Table 1.1 Parameters of SimBlock
From Aoki *et al.* (2019)

Parameters	**Description**
Average mining power	Average mining power (hash rate) of each node block
Block interval	Expected time taken to generate a block
Block size	Maximum size of a generated block
Distribution of degree	Cumulative distribution of number of outbound links
Distribution of region	Distribution of node's region
List of download bandwidth	List of download bandwidth assigned to each region
List of latency	List of latency assigned to each region
List of region	Regions where nodes are situated
List of upload bandwidth	List of upload bandwidth
Max block height	Block height when a simulation end
Number of nodes	Number of nodes participating in the blockchain network
Routing table	Types of routing tables
Standard deviation of mining power	Standard deviation of mining power (hash rate)

VIBES, a blockchain simulator that models large peer-to-peer networks, is described in (Stoykov, Zhang & Jacobsen, 2017). In addition, it does not adhere to the Bitcoin protocol and is capable of supporting large-scale simulations involving thousands of nodes. A fast forwarding concept is used in this simulator to improve its scalability. Using relevant input parameters and empirical and theoretical results, the simulator can therefore skip the heavy computations.

Alharby and Moorsel introduced another blockchain simulator in 2018 (Alharby & Van Moorsel, 2019). In order to implement this simulator, three layers were considered: incentive layer (decision making mechanisms of stakeholders), connector layer (consensus mechanisms) and system layer (network configuration). As part of the block creation process in this simulator, the Proof of Work algorithm is implemented.

The (Gervais *et al.*, 2016) simulator was developed to evaluate the durability of blockchain networks against double spoofing attacks based on parameters such as the block size and block interval. As shown in Table 1.2, the simulator was demonstrated with the following parameters.

Table 1.2 Parameters used by simulator
From Gervais *et al.* (2016)

Parameter	**Bitcoin**	**Litecoin**	**Dogecoin**
Block interval	10 mins	2 mins 30 secs	1 min
Average block size	534.8KB	6.11KB	8KB
Number of public nodes	6000	900	600
Number of connection	Distribution based on (Miller *et al.*, 2015)		
Geographical distribution	Distribution based on actual blockchains		
Bandwidth and propagation delay	6 regional bandwidth and propagation delay		

The simulator introduced in (Faria & Correia, 2019) has the same name as (Alharby & Van Moorsel, 2019), Blocksim. For the purpose of simulating a blockchain model and measuring its parameters, they have modeled random phenomena in order to mimic the behavior of a real system. As an example, to determine the throughput of sending and receiving messages between two nodes of a P2P network over 24 hours, they collected different log files from the nodes for 24 hours and found the most appropriate probability distribution function which fits the collected data. As well as latency, transaction validation and block validation times, the same strategy is employed for other parameters. By sampling from a probability distribution function, the simulator simulates random phenomena following a stochastic simulation.

In (Pandey, Ojha, Shrestha & Kumar, 2019), a comprehensive and open-source simulation tool for blockchain systems is presented. The simulator can assist blockchain architects in better evaluating the performance of planned private blockchain networks by running scenarios and determining the optimal system parameters.

1.2 Blockchain behavior analysis

In (Aoki & Shudo, 2019), the authors attempt to improve the network topology of the nodes and shorten the propagation time to shorten the block generation interval. They propose a method for selecting neighbors that will form a network topology that will propagate blocks rapidly. Their method involves each node evaluating the other nodes using a scoring scheme in order to identify the appropriate node to send the block generation announcement. In spite of the fact that their results showed an improvement in block generation time, they did not provide information regarding the fork rate. This is an extremely critical aspect.

There is a model presented in (Liu, Qin & Chu, 2019a) that reduces the block propagation delay, which is the major cause of forking. Their method involves the recipient node deciding whether to validate the newly received block or accept it without validation when it receives a block. In order to accomplish this, they have developed a probabilistic verification process. During this process, a random number is generated by the node when it receives a new block. The node accepts the block without validation if the number is greater than the validation degree. According to the authors, their proposed model, PvScheme, can result in a minor delay in block propagation and fewer forks. By defining some additional factors to ensure reliable block delivery, this study also enhances the security of PvSchIn this study, PvScheme is evaluated in terms of its resistance to double-spending attacks and fake blocks. There is a major drawback to their method, which is that it considers propagation delay as the only cause of forking. This is despite the fact that it is affected by a number of factors.

The model proposed in (Shahsavari, Zhang & Talhi, 2019a) is implemented in the network simulator OMNet++. The model uses an Erdos-Renyi random graph to model the delay and traffic overhead of a Bitcoin network. Based on the block size and number of connections per node, it derives key features of the Bitcoin network's performance. A comparison has been made between the simulation results and the real data collected from the Bitcoin network in order to validate their results. During their experiments, they found that increasing the block size increased Bitcoin throughput and this resulted in a significant increase in block propagation time. Even though this delay can be reduced by increasing the number of connections per node, it will result in an increase in network traffic overhead.

The authors extend their work by presenting a theoretical model for fork analysis (Shahsavari, Zhang & Talhi, 2019b). In this paper, Erdos-Renyi random graph construction of the overlay network of peers has been developed as a theoretical formula to model Bitcoin consensus and network protocols. With the help of this model, the researchers were able to demonstrate that network bandwidth, block propagation delay, and block size have a dramatic effect on the occurrence of forks. Additionally, the results showed that there is an inverse relationship between the block time and the probability of a fork endangering the security of the blockchain.

This study employs a simulated network containing nodes distributed across a variety of geographical regions to examine the performance of block propagation in (Mišić, Mišić & Chang, 2019). It was also attempted to analyze the effect of different parameters on the fork rate. The authors demonstrate in their experiments that the mean round-trip time is the primary cause of block propagation latency, which means that forks occur more frequently when the mean round-trip time is longer.

Simulators have been used in a number of studies to evaluate novel blockchain designs. In addition, researchers studied blockchain behavior, particularly fork rates. It is critical to note that the fork is affected by several parameters, but only a few have been considered. Further, although many researchers have examined the implications of EIP-1559 on blockchain behavior, users, and miners' benefits, a comprehensive analysis of the fork issue has not yet been conducted. The effectiveness of forks is evaluated based on a variety of effective parameters. The next section provides a detailed description of the evaluation function and its input parameters.

1.3 Sharding

According to (Yun, Goh & Chung, 2019), malicious nodes can take over a shard and compromise the entire network. Shards are more susceptible to 51% attacks because they compute at a fraction of the network's speed. They propose calculating the trust score of all nodes based on consensus results in order to prevent collusion between malicious nodes. Afterward, a genetic algorithm is used to calculate the distribution of nodes.

According to (Cai *et al.*, 2021), the paper proposed a shard validation validity model that considers four objective factors, including shard invalidation probability, delay, throughput, and malicious node load. Using a dynamic reward and penalty mechanism, they attempted to solve this multi-objective optimization problem. Malicious nodes were minimized to prevent them from aggregating. According to the proposed method, a blockchain-enabled IoT environment can resolve the conflict between throughput and shard validity for enhanced security.

In order to achieve optimal throughput and security, (Yun, Goh & Chung, 2020) has designed a sharded blockchain using deep reinforcement learning. The status of the network is analyzed using an artificial intelligence model. A number of parameters were considered in the simulation, including the maximum number of shards, the maximum block interval, the average transaction size, and the number of nodes. The deep learning agent is capable of determining the block size, block interval, and number of shards.

ChainsFL is a federated learning framework based on a sharded blockchain introduced in (Yuan, Cao, Peng & Sun, 2021). A sharded blockchain architecture based on Raft has been developed by them in order to improve the scalability of their framework. Comparatively to similar methods, ChainsFL provided a higher convergence rate when training convolutional neural networks.

To enhance the scalability and improve the performance of the network, Yoo and Daejeon presented the blockchain framework as part of a domain-based static sharding framework (Yoo, Yim & Kim, 2018). As part of the PBFT process, each shard's transactions are validated by a committee. It is expected that blockchains will be more effective as a result of their framework.

The authors of (Nguyen, Nguyen, Dinh & Thai, 2019) proposed an algorithm called OptChain that optimizes transaction placement between shards to reduce cross-shard transaction traffic. Compared to OmniLedger, OptChain reduces latency by 93% and increases throughput by 50%.

A hybrid method for assigning nodes to shards is proposed in (Bugday, Ozsoy & Sever, 2019) that combines a machine learning algorithm with a VRF (Verifiable Random Function) function. It is one of the primary objectives of this method to reduce the likelihood of shards and blockchains deteriorating.

There is one major drawback to all previous research on blockchain technology: no one has examined the impact of sharding on the likelihood of forks occurring. The main objective of this research is to overcome this drawback by analyzing the likelihood of forks occurring in sharded networks. In the following section, the experiments and results are described in more detail.

1.4 Governance methods

In (Arribas, Arroyo & Reshef Kera, 2020), CLAUDIA is a creative approach that combines on-chain and off-chain governance. By using this method, stakeholders will be able to discuss various issues and track issues on the blockchain using on-chain services such as Ethereum-based DAOs like WUDDER, in addition to an off-chain governance compliance desk. It is possible to resolve issues efficiently by utilizing both off-chain and on-chain discussions. CLUADIA offers a sandbox for testing upcoming features for limited users in addition to being more user friendly. As a result, it is easier to integrate into existing business models and comply with regulatory requirements. Meanwhile, it is vulnerable to the inherent limitations of the Ethereum blockchain due to its reliance on the Ethereum network.

In order to protect against a Sibyl attack, many current blockchain technologies use coin-based voting schemes. According to (Chung, Nair, Ravi & Kajgaonkar, 2021), proof of participation would be an effective method for addressing these deficiencies. As a proxy for an individual's identity, a user's participation and level are used during a decentralized crypto game. It helps to prevent sybil attacks while also distributing voting power among a variety of stakeholders.

According to (Li & Zhou, 2021), the author discusses not only the governance of blockchain technology, but also how it can be incorporated into a framework for blockchain governance. Blockchain technology plays a significant role in blockchain regulation under this framework.

Blockchain ecosystems may benefit from the integration of governance of blockchains with governance on blockchains.

To balance block composition time and transmission time, the authors used multi-objective Particle Swarm Optimization algorithms (PSO) in (Singh & Vardhan, 2020). Based on their findings, 213 transactions per block is the appropriate block size. Although they have utilized optimization methods to control block creation and transmission time, their method does not constitute an on-chain governance model.

Another study utilized multi-objective PSO to solve a block size optimization problem in (Singh & Vardhan, 2021). The objective functions considered were the time required to build blocks and the time required to select transactions. According to the results, a block size of 3.8 MB optimizes the time spent on the selection of transactions and the construction of blocks.

1.4.1 Machine learning-based blockchain consensus

In order to train an accurate machine learning system, it is necessary to have access to a large dataset and to have a large amount of computing power. Today, most of it is centralized to large corporations such as Google, Amazon, etc. In order to facilitate fair access to learned models and create an open repository, a distributed method has been introduced, which can be found in (Bravo-Marquez *et al.*, 2019). (Bravo-Marquez *et al.*, 2019) proposes a new consensus mechanism for the cryptocurrency WekaCoin based on ranking machine learning model training for a given task. It is called Proof-of-Learning and is used for models of supervised learning. Also, it aims to replace the conventional proof-of-work consensus with a proof-of-useful work with respect to the training schema (Bravo-Marquez *et al.*, 2019).

There are three categories of nodes in this type of network, including miners, suppliers, and trainers. It is the suppliers' responsibility to provide trainers with a learning task that consists of a desirable threshold for them to learn and propose the updated model to the network. Using the ranking schema generated by random validators, the best model owner and validators will be rewarded with newly mined tokens and all trained models will be stored in a distributed ledger

(Bravo-Marquez *et al.*, 2019). A block contains information regarding the list of transactions, previous block hash, and metadata regarding the learning competition, and there are three types of transactions: standard transactions, task publication transactions, and model transactions through the proposed model (Bravo-Marquez *et al.*, 2019). The proposed model architecture is as follows:

When a supplier participates as a trainer in a learning competition, one of the most likely instances of malicious behavior can occur. In view of the fact that validators are selected at random, a majority control should be maintained on the network in order to accomplish this. Moreover, to prevent DOS attack and Sybil attack, trainers should pay a small fee to propose trained models as model transactions in network (Bravo-Marquez *et al.*, 2019). In the same manner as the File Coin protocol, committees are selected based on proof-of-storage, and block proposals will be made by a select committee so that a consensus can be reached on three main items for each block, including the transaction list, the method used for validating a block, and the nested model for the chosen method. A continuous supply of tasks and collusion among three key actors are the major problems (Bravo-Marquez *et al.*, 2019). Submission deadlines, performance metrics, rewards, and a ranking schema are the minimum requirements that we take into account when evaluating a machine learning competition.

There is a possibility that the supplier may engage in malicious behavior when he participates as a trainer in his learning competition. In order to select validators, a majority control needs to be established on the network, since it is a random process. It is also recommended that trainers pay a small fee to propose a trained model as a transaction in the network in order to prevent DOS attacks and Sybil attacks.

In contrast to this research, our proposed framework ensures that a continuous supply of learning tasks is provided in order to continuously modify the blockchain network and optimize its performance. Our protocol differs from others in that the learner nodes are considered to solve the multi-objective optimization problem of the blockchain network during the learning phase.

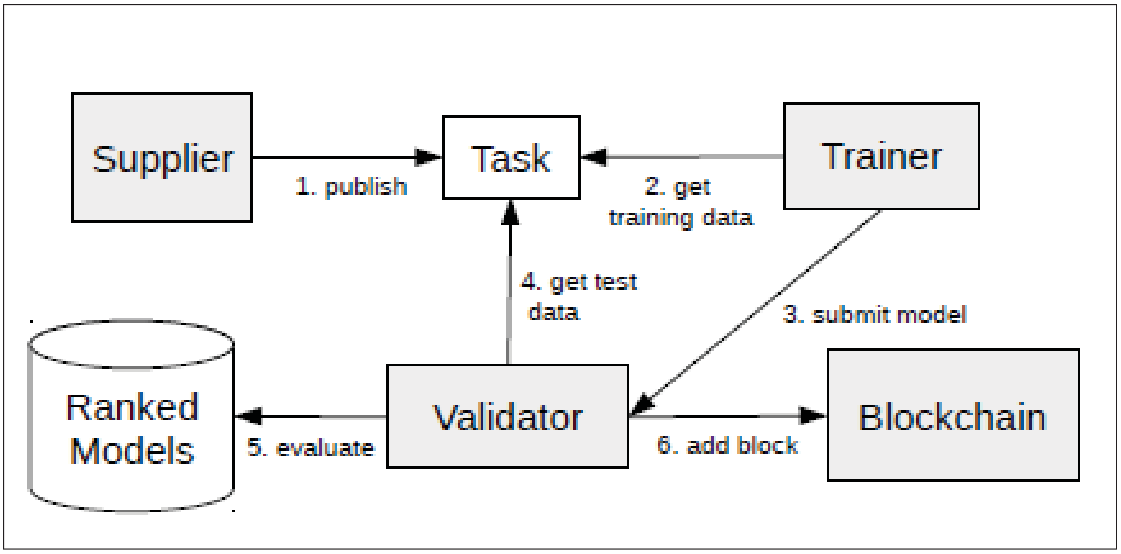


Figure 1.1 Proof of learning

1.4.2 Solving optimization problems for blockchain consensus

(Shibata, 2019a) addresses the huge amount of energy wasted in the Bitcoin network to solve a cryptographic puzzle as the main problem and proposes a new consensus model known as Proof-Of-Search (POS), under which miners compete for an accurate solution to any optimization problem (Shibata, 2019a). A blockchain of this type includes some built-in functionality for submitting and receiving optimization tasks through the network, as well as a POW-type job that is used whenever there are no jobs available. According to this study, the probability of fork occurrence and variance of block time are lower than those of conventional POW models (Shibata, 2019a). In the proposed network, there are different terms such as searcher, evaluator, and client, and collusion between the client and miner will be taken into account. Evaluators should comply with the leading zero requirement in order to generate a valid hash that is concatenated from a solution and its evaluation in the same manner as Bitcoin. There are two objectives for any evaluator: finding a good nonce and a good solution for the optimization model (Shibata, 2019a).

The reward will be paid following the addition of a block to the blockchain, and each node will verify that its solution is genuine. It also proposes the use of mini-blocks to consider the winning probability of any miner in relation to the computation time of each job (Shibata, 2019a). Below is a description of the architecture:

Miners would be able to share their internal evaluations and reuse them if they were able to find a good evaluation rather than a promised hash value. Any evaluation is associated with the miner ID in order to prevent this from occurring. To incentivize the miners, the clients charge them in native coins to find a good approximate match for the submitted job (Shibata, 2019a). We propose a framework that ensures a decrease in fork occurrence compared to this work, since the learner nodes participate in the blockchain optimization problem during the learning phase and are not required to participate in the mining phase or transaction validation phase.

In order to demonstrate that the dynamic design of a blockchain leads to a stable and fork-free distributed ledger, this is a primary metric to prove that the fork possibility will decrease, and this work has direct implications for our modeling of fork occurrences. The proposal proposes a dynamic reward based on the required computation for any job based on checking the past block times, leading zeros, and charges in the same manner to prevent the occurrence of a fork in terms of halving the block rewards. We should also take into account another factor when counting the computation steps when it may lead to a denial of service attack.

Additionally, in (Chen, Duan, Zhang, Zeng & Wang, 2018a), neural networks (NN) are used to support consensus protocols in order to ensure decentralization and safety of the blockchain network, and the number of supernodes is determined based on a dynamic threshold, enabling them to validate the upcoming blocks. Additionally, it utilizes the Conventional POW, POS, and DPOS consensus protocols, and proposes Proof-Of-AI as a new consensus protocol during the paper (Chen *et al.*, 2018a). Supernodes have greater computational power and less network latency, which will be taken into consideration for PoAI, while random nodes are considered to ensure the fairness of the blockchain network (Chen *et al.*, 2018a). The following pictures illustrate the node pool:

Accordingly, it calculates Average Transaction Numbers (ATN) using nodes' characteristics such as computing power, connection numbers, and online time, and then uses a Convolutional

Neural Network to train ATN (Chen *et al.*, 2018a). Also, it provides a method of selecting random nodes as supernodes which has an impact on our research in which we can use this method in our model in a public blockchain network for ordinary miners as well as super miners to learn the parameters instead of all miners having to go through the same learning process.

It addressed the huge wasted energy in the Bitcoin network to solve a cryptographic puzzle as the main problem and proposed a new consensus model called Proof-Of-Search (POS), in which miners compete to find the best solution to any optimization problem. This research employs a POW-like job that is used whenever there is no job to submit, and some built-in functionality for submitting and receiving optimization tasks through the network. This model asserts that the probability of forks occurrence and the variance in the block time are lower than those of conventional Proof-of-Work (POW) models.

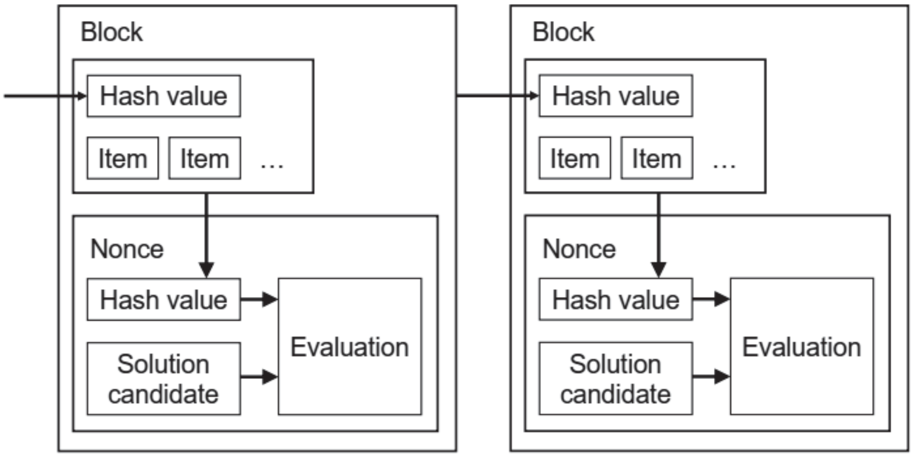


Fig. 2: Distributed timestamp server in a minimal PoS scheme

Figure 1.2 Distributed timestamp

While this work does not provide a theoretical formulation of the decentralization and performance of the blockchain network, we propose a mathematical formulation for the efficient configuration finding problem as an optimization problem. Additionally, our proposed framework incurs less overhead on the blockchain network and its processes. Our proposal, in contrast to this work, takes into account an on-chain reputation mechanism for the learner, which serves as a feedback mechanism for selecting the most suitable learner throughout the learning process.

It is the client's responsibility to pay the miners with native coins in order to incentivize them to find the best possible approximation of the job submitted. In order to receive the reward, a block must be added to the blockchain and each node must check whether the solution is genuine. In addition, mini-blocks are proposed to consider a miner's winning probability in relation to its computation time, in which each mini-block is corresponding to a specific task. As shown in Figure 1.2, the architecture is as follows:

1.4.3 Public blockchain and federated learning

A centralized aggregator plays a crucial role in federated learning in terms of storing and updating global models. A decentralized aggregator is to be replaced by an on-chain aggregator in this method (Ramanan, Nakayama & Sharma, 2019). The network provides them with an optimization problem in order to update the global model. In order to reach consensus, they must solve the optimization problem. We can leverage federated learning through blockchain networks to update the ultimate models in blockchain based aggregators using federated learning (Ramanan *et al.*, 2019).

As an alternative to the central server, participants of the public blockchain are able to share their local model updates while verifying as well as remuneration mechanisms. In doing so, they will be able to include a variety of untrustworthy participants and this will result in greater accuracy when training samples and participants are large (Ramanan *et al.*, 2019).

An incentive mechanism and a fair platform are required to facilitate the commercialization of federated learning. Privacy protection and the effectiveness of collaborative modeling among participants should be taken into consideration during implementation of the program. Participants who contribute more data should be rewarded and incentives should be implemented through a consensus mechanism (Ramanan *et al.*, 2019). This will enable them to participate and own their profits regardless of the size of the data they have. Lastly, data samples must be kept confidential from other participants. The consensus protocol and incentive mechanism are the two major standards that we need to develop in order to use blockchain networks for federated learning.

Based on the proof-of-work consensus mechanism, the authors developed blockchain-based federated learning (BlockFL) (Kim, Park, Bennis & Kim, 2019), which focused on analyzing end-to-end latency. In analyzing the learning performance of the proposed blockchain-assisted decentralized FL (BLADE-FL) (Ma *et al.*, 2022), the issue of single points of failure was addressed. Through the integration of asynchronous global aggregation, the federated learning with asynchronous convergence (FedAC) method proposed by (Liu, Qu, Xu, Hao & Gu, 2021a) is designed to enhance communication performance. Despite focusing on the performance of various system components, these studies fail to address the scalability requirements of large numbers of participants. Furthermore, they do not consider the impact of the consensus mechanism on throughput. Additionally, blockchain-based federated learning applications have been proposed, along with domain-specific frameworks, such as healthcare (Passerat-Palmbach *et al.*, 2019) and IoT (Zhao *et al.*, 2020). These works emphasize the plausibility of cutting-edge implementations rather than their scalability or performance. In accordance with (Gai, Wu, Zhu, Zhang & Qiu, 2019), it is possible to develop a cloud-based task allocation system which preserves privacy. Using blockchain technology enhances the trustworthiness of edge nodes and the security of communication networks.

As part of our work, we have also considered an aggregator in order to determine which configuration set proposed by the learner is the most suitable. As part of this study, we considered two different types of nodes, namely "miners" and "learners". The network optimization is formulated as a multi-objective optimization problem in which the inputs are derived from the local parameters of the learners. Finally, we have also included a deposit schema in our solution to ensure that learner nodes act honestly. In our proposed model, the separation of consensus mechanisms can be used to increase throughput.

1.4.4 Decentralized AI on blockchain

In the last decade, blockchain technology has become one of the most disruptive technologies, and its integration with AI is expected to provide more opportunities for our world (Dinh & Thai, 2018). It has been demonstrated that AI enables blockchains to have a better performance in consensus mechanisms by using supervised learning algorithms to detect anomalies (Chen, Ji, Luo, Liao & Li, 2018b). A majority of machine learning models are currently centralized, predictions are sold per query, and published models can quickly become out of date if they are not retrained. In this work, participants will be able to collaboratively build a data set and utilize smart contracts to host a continuously updated model. The models will be free to infer (because reading Ethereum doesn't charge gas) as well as a free set of data stored on the blockchain (models will be stored in smart contracts) (Nasir, Qasse, Talib & Nassif, 2018).

(Harris & Waggoner, 2019) presents a framework for sharing and improving machine learning models, focusing on supervised learning with a particular emphasis on incremental learning models based on one sample. This approach is used for training a single layer perceptron model using IMDB reviews data sets. Three customizable components are included, including an incentive mechanism, a data handler, and a machine learning model, as well as on-chain updating of models and off-chain prediction (Harris & Waggoner, 2019).

In terms of incentive mechanisms, gamification is the first contribution, which rewards participants with points and badges for providing good data. An outside party (academic or company) provides a pool of reward funds and test data for the purpose of empowering. A deposit, refund, and take scenario is another incentive. It requires a deposit when contributing data, and the refund is subject to a time limit, so you may claim and take all of the deposit based on the actions of other participants (Harris & Waggoner, 2019). The proposed architecture for the model is as below:

The majority of our considerations were learning tasks requiring a protocol upgrade. The purpose of this is to determine the best parameters of the protocol based on the given history, which can be viewed as a learning challenge. Our contribution to handling potential issues would be to consider the learning task execution in a decentralized manner, so the model and training model would be useful.

1.5 Adaptive blockchain systems

In recent years, blockchains have been designed to be adaptive. A blockchain with improved throughput has been proposed by the authors of (Qiu *et al.*, 2018). A dynamic adjustment is made to the access nodes, the primary nodes, and the computing capability of the blockchain. As a result, the blockchain was able to significantly increase its throughput (Qiu, Ren, Cao & Mai, 2020).

In distributed ledger technology (DLT), a virtualization approach is considered based on (Yu, Liu, He, Si & Zhang, 2018). It is important to note that logical resources abstract the underlying resources, such as hardware, compute, storage, and networks. To improve performance, these resources are scheduled according to a flexible schedule (Qiu *et al.*, 2020).

Though the current inflexibility of many dominant public blockchains, such as Bitcoin, has been widely criticized (Iwamura, Kitamura, Matsumoto & Saito, 2019; Courtois, Grajek & Naik, 2013), it is still unclear how proof-of-work blockchain networks can be configured to be more resilient to external disruption.

Recent efforts have been made to view blockchain networks as adaptive systems. Several studies, such as (Fullmer & Morse, 2018; Hovland & Kucera, 2017), have not considered reward as a variable for improving difficulty adaptation. Nevertheless, in (Saito & Iwamura, 2019) address reward adaptation for the purpose of price stability: the network inflates or deflates in response to changes in demand.

The impact of that work on power and, therefore, security is not the focus of that work. According to (John & Pam, 2018), blockchain networks can be described as complex adaptive systems. This describes agent behavior in response to local signals left by other agents. Based on the findings of this study, it may be useful to understand how configuration decisions are acquired, communicated, and enforced within blockchain networks.

In addition, (Zargham, Zhang & Preciado, 2018) propose a linear representation of blockchain networks derived directly from control engineering. While the model has not been specifically designed for the configuration variables we discussed here, it may be useful in this context.

In contrast, (Lin *et al.*, 2018) propose a mechanism for determining the transaction fee rate based on past transaction volumes. We have objectives that are somewhat different from those of that work, namely, the sustainability of proof-of-work in the absence of mining rewards, as will ultimately be the case with Bitcoin.

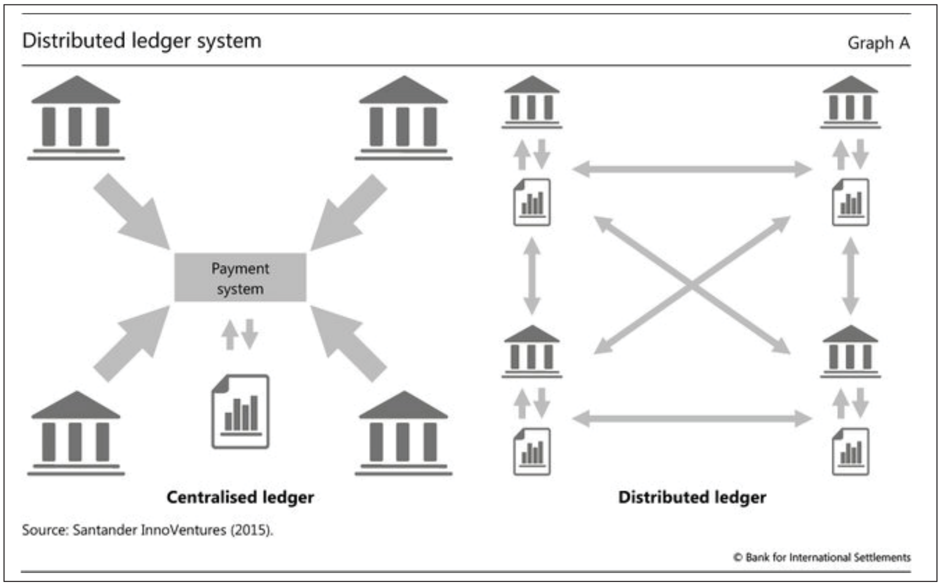
CHAPTER 2

BACKGROUND

In this section, we provide background information that is necessary for a better understanding of our current work. An extended version of the fork probability model developed by the previous studies can be found in this section, along with a detailed description of EIP-1559 and Blocksim, the simulator on which our implementation is based.

2.1 Blockchain technology

Based on the Figure 2.1, centralized and decentralized architectures are compared in terms of how they organize their work (Μακράκης, 2018). A centralized architecture involves the participation of a third party (banks), while a decentralized architecture involves the participation of the participants directly without the involvement of the third party. Each participant has the ability to observe and store the information, which is why this is important. It is through this transparency that the decentralized architectures become safer and more trustworthy.


Distributed ledger system
Graph A
Payment system
Centralised ledger
Distributed ledger
© Bank for International Settlements

Source: Santander InnoVentures (2015).

Figure 2.1 Centralized architecture vs. Decentralized architecture

The first attribute of the blockchain is that it operates through a peer-to-peer network that shares information on the insertion of new transactions into new blocks and the addition of new blocks. Secondly, dissemination of information is usually accomplished through flooding, as opposed to peer-to-peer overlays, which are constructed through some form of peer discovery. Lastly, cryptocurrencies, as well as other crypto assets, must be technology-ready in order to become widely accepted and adopted. To achieve this goal, a transaction layer will be required, which will be essential for gaining the most comprehensive adoption possible.

As illustrated in the Figure 2.2, the blockchain is composed of a hierarchy of layers, each one representing a specific aspect of the blockchain and can be thought of as a protocol stack (Rosa *et al.*, 2019). A more detailed description indicates there are at least three layers above the Internet layer.

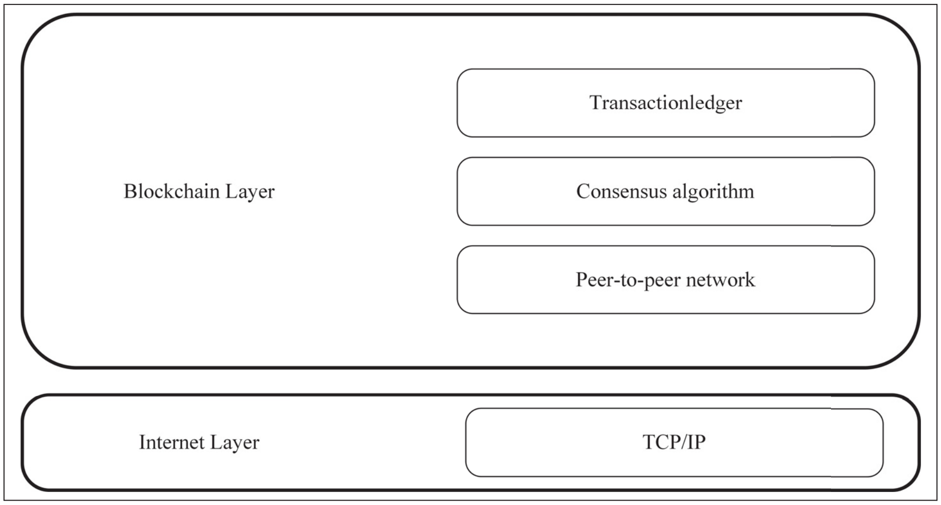


Figure 2.2 Blockchain layers
From Rosa *et al.* (2019)

As a result of the layered structure of the blockchain, it is possible to isolate and understand the protocols of the various components more effectively, but it is evident that the effectiveness of each layer is greatly dependent upon its function in the other layers. Consequently, it is important to evaluate the various alternatives to each component as well as the effects of any potential modifications. Even so, the complexity of the technology and the wide-scale nature of the distributed system make the evaluation process extremely challenging. Therefore, it is important to simulate the blockchain as part of the evaluation process.

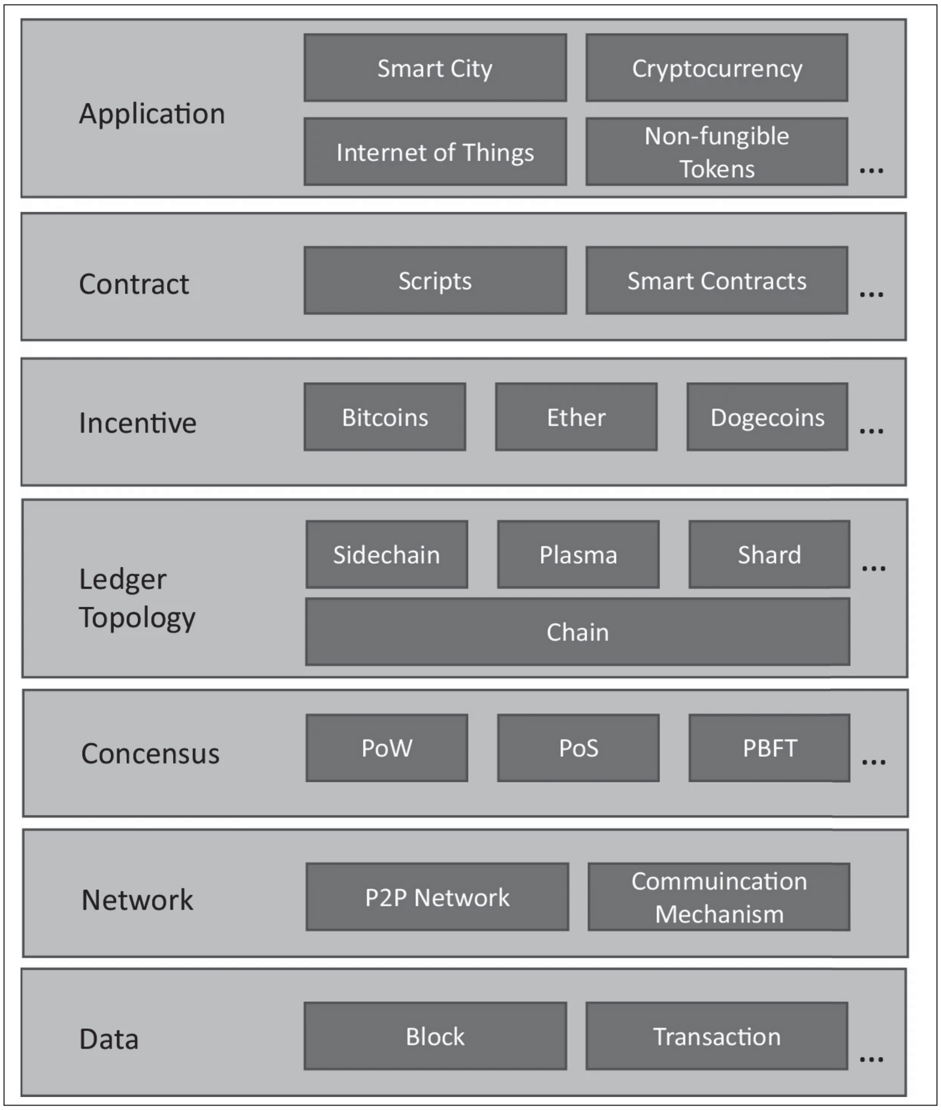


Figure 2.3 Decomposition of the structure of blockchain
From Nartey *et al.* (2022)

For a better understanding of the structure and nature of Blockchain systems, it is helpful to examine the layers of Blockchain, as shown in 2.3. Described in the Figure are the data, network consensus, ledger topology, and contract and application layers (Yang, Yu, Si, Yang & Zhang, 2019). The data layer encapsulates and verifies data, ensuring the privacy and security of the data generated during a Blockchain application or transaction. The header of this block contains a hash value that identifies it as a follower of the previous block. A chain of blocks is generated by this process, which is replicated on all nodes of the Blockchain (Li *et al.*, 2020a). As part of the replication process, generated blocks are propagated to all nodes of the Blockchain (Cai *et al.*, 2021).

2.1.1 Consensus

A consensus algorithm is used in order to ensure that all nodes are in agreement regarding the evolution of the blockchain. The Bitcoin protocol uses a random selection algorithm, while the Ethereum protocol uses the Kademlia node discovery protocol (Alharbi & Hussain, 2015). The Bitcoin blockchain is based on a Proof-of-Work consensus scheme. In addition to Proof-of-Stake, there are a number of other options, including Proof-of-Authority (Mingxiao, Xiaofeng, Zhe, Xiangwei & Qijun, 2017). Data and transactions are recorded in the transaction ledger, which is located above the consensus layer. These technologies have made it possible to develop smart contracts executed on the blockchain, thanks to blockchain 2.0, based on Ethereum.

Based on a combination of random selection and wealth or age, proof-of-stake is a consensus protocol that determines the creator of the next block. It was initially implemented in Peercoin (Popper, 2013). As a result of this protocol, the node with the most coins will be able to create blocks more frequently. This will result in a greater number of coins being granted. Trust is required for the implementation of this protocol. It is more likely that the most wealthy nodes will be selected, and therefore they will have control over the network.

Proof-of-activity (Bentov, Lee, Mizrahi & Rosenfeld, 2014) combines the advantages of proof-of-work and proof-of-stake. As part of this scheme, miners add an empty block header to the block using a PoW algorithm. In the same manner as proof-of-stake, the header specifies a random group of validators. It is necessary for these validators to sign the newly created block. In order for a new block to be added to the chain, it must be signed by all of the selected validators. Proof-of-activity has the advantage that it requires both a majority of CPU power and a majority of coins to control the cryptocurrency (Shibata, 2019b).

A proof-of-space (Dziembowski, Faust, Kolmogorov & Pietrzak, 2015) protocol involves a prover and a verifier that store large amounts of information. It is necessary for the verifier to request a piece of data from the prover in order to verify that the prover is still storing the data. This system reduces the complexity of computing, storing, and communicating with the verifier. Using proof-of-space in a decentralized blockchain requires a method of determining the winning node and an indication of the likelihood that each miner will win. There should be a correlation between the amount of data stored on each node and the probability of winning. You can find a discussion of these practical considerations at (Park *et al.*, 2018). As discussed in the paper, proof-of-space has several weaknesses. One of the problems is the capability of nodes to mine on multiple chains simultaneously. The miners may also attempt to create many different blocks by altering the block contents slightly and announcing the most favorable block with a single proof-of-space (Shibata, 2019b).

The increased speed of DPoS leads to a higher level of security than the original PoS consensus model. The delegation of block producers has also been identified as a form of democracy within the blockchain, with different coin holders voting for the delegation. The DPoS model also offers much faster transaction processing times than PoW, as both PoS and DPoS suffer from different types of security challenges, including a higher degree of decentralization (He, Tang & Wang, 2020).

Delegates are elected by other users to vote on their behalf in DPoS. As a result, it is the voters who have the power. If a voter witness fails to perform or misrepresents their vote, they may be removed from office. Delegates are responsible for distributing the benefits they receive to the voters who elected them. Users are restricted from participating in the validation process. It is possible for users to select individuals to represent them, however, the delegates may abuse their power. Due to their responsibility and authority, they are responsible for validating blockchains.

As a result of the presence of cartels in the system, it is more vulnerable to attacks and less decentralized (Saad, Radzi & Othman, 2021).

2.1.2 How blockchain works

As a result of the lack of central authority, for example, if Alice wishes to send some tokens to Bob, she must broadcast a message to the network announcing the number of tokens being transferred to Bob's account address. As a participant in the network, each node receives the message, updates its copy of the database, and passes the message on to its neighbors. After that, the miners begin the validation process.

Consequently, in this situation, some conditions must be verified, including Alice's claim amount and Bob's account address. A proof of work is prepared by the miner nodes by solving a mathematical puzzle. Upon obtaining a valid proof of work, a miner will be rewarded by the network with some tokens. The solved puzzle, along with the requested information, will be added to all copies of ledgers. The miners perform this process for a batch of requests rather than doing it one at a time, which is known as a block (Gervais *et al.*, 2016).

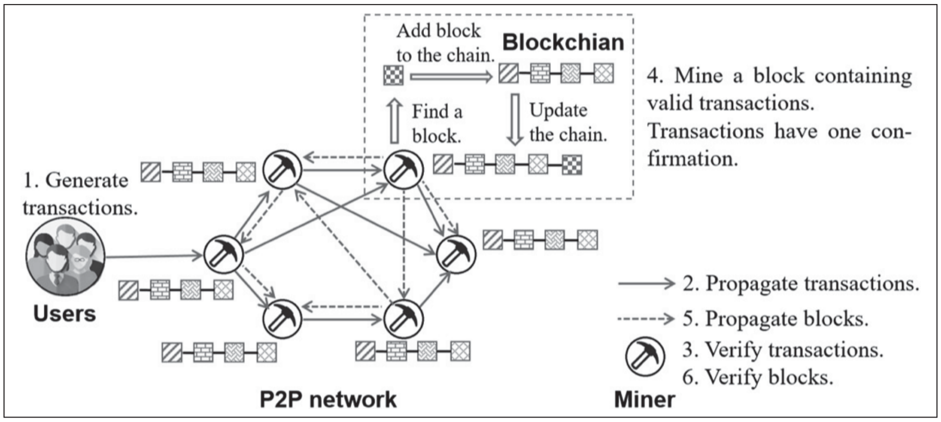


Figure 2.4 The processes of transactions in blockchain
From Zhang *et al.* (2021a)

Blockchain mining involves validating and appending the latest transactions to the ledger. As transactions are conducted on a blockchain, miners validate them and add them to the blockchain as blocks. It is imperative that miners solve a mathematical puzzle in order to ensure that their computing power is invested in a block that is accepted by the network. As the puzzle difficulty changes over time, the first miner who solves the puzzle and propagates the block through the network will be rewarded. Once the majority of nodes have reached a consensus and approved the solution, the puzzle will be resolved. As soon as the most recent block is added to the blockchain, all copies of the blockchain will be updated. As a minimum, the following processes are required for the operation of a blockchain.

1. Establishing a network or connecting to one. Blockchain technology is based on a process called network formation: finding and joining networks.
2. Memory pool. A temporary placeholder or shared space is available to the entire network for accumulating transactions, which are then picked up by miners for mining (Atik & Gerro, 2018). Unconfirmed transactions are those that are awaiting confirmation. A limited number of transactions with an accumulated size of 1MB are selected by the Bitcoin miner every 10 minutes for mining (McGinn *et al.*, 2016).
3. Handling transactions. In our context, a transaction refers to an external instruction performed on a blockchain platform. As an example, in Bitcoin, a transaction is an instruction to pay a certain amount of Bitcoins (Sunyaev, 2020). It is possible for a transaction in Ethereum to be the execution of a smart contract (Mohammed, Abdulateef & Abdulateef, 2021). The method of handling transactions varies from platform to platform. However, in the vast majority of blockchains, once a node has internally processed a transaction, it is transmitted to the peers. Nodes validate and store transactions in their memory pools as soon as they reach them. Memory pools are used by nodes to provide a list of unconfirmed transactions that will be included in upcoming blocks (Golosova & Romanovs, 2018).
4. Forming blocks. In a blockchain, transactions are collected into blocks. Blocks are created for all pending transactions in a miner's memory pool. These transactions transactions are chained to a reverse linked list data structure using cryptography so that they cannot be modified. Once a block has been created, it is broadcast to all peers connected to the network.

5. Block commitment. A consensus algorithm is used to validate each block before it is added to the blockchain. According to consensus rules, the block, its meta-data, individual transactions contained within the block, and specific information contained within the block, such as a timestamp, must be validated. As soon as the validation process has been completed, the validated blocks are added to the local copy of the blockchain that resides at the node. In this way, consensus algorithms ensure that ultimately, within certain limits, networks will be consistent.

2.1.3 Block and transaction propagation

In a blockchain, information pertaining to transactions is stored in a series of chained blocks. In a distributed, peer-to-peer network, each node maintains a local copy of the entire blockchain in addition to propagating, validating, and storing transactions. There are generally two types of nodes: non-miner nodes and miner nodes. Whenever a non-miner node creates a transaction, it sends it to its neighbors for verification and broadcast. A miner node chooses transactions based on their suggested fees (the higher the fee, the greater the chance that the transaction will be chosen by the miner), while taking into consideration block size limitations and solving a mathematical puzzle called consensus algorithm (such as proof-of-work) in order to produce a random hash; the primary identifier of a block is a cryptographic hash generated by the SHA256 algorithm (Gueron, Johnson & Walker, 2011).

A miner adds a newly mined block to the local copy of the ledger and propagates it to the network by broadcasting the '*new block*' message to neighbors. In order to join the chain, the recipient nodes ask the miner to send them the header of the new block using the message '*get-header*'. As a final step, the miner responds with the message '*header*' and sends the block header. Three scenarios are considered when checking block hashes:

1. 1- if the previous hash of the new block is equal to the hash of the last block of their local chain, the recipient nodes add it to their local chain.
2. 2- if the previous hash of the new block is not equal to the hash of the last block but it is equal to the hash of an older block, the new block is prone to create an accidental fork.

3. 3- if the previous hash of the new block is not seen in any of the previous blocks, the recipient stores this block in the parent queue to receive the child block sooner than the parent block due to network delay.

There is a difference in the modeling of Ethereum block propagation and Bitcoin block propagation. As shown in Figure 2.5, Node A sends an inv message to its neighbours in order to announce the creation of a new block. Node B receives this inv message and calls Node A using a getdata message to receive the entire block that was announced by Node A. In response, Node A transmits the entire block to Node B using a block message. Upon receiving this message, Node B validates the block and adds it to the chain (if it is valid). Nodes on the network process new transactions in the same manner as those announced by nodes. The miner node adds new transactions to the transaction queue when it gets a transaction message (Faria & Correia, 2019).

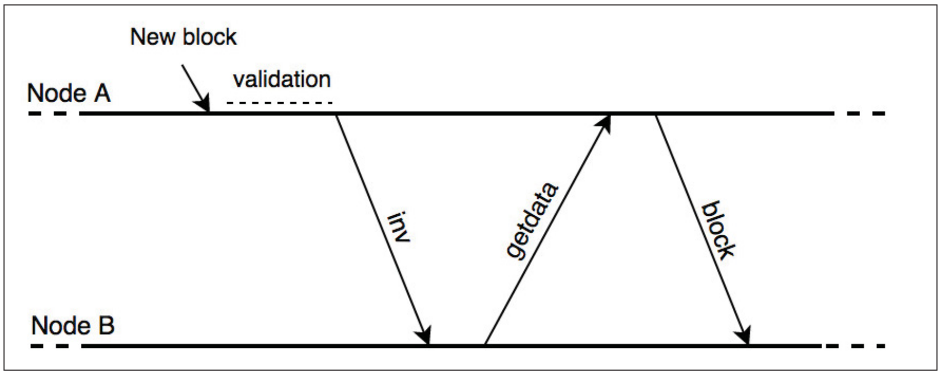


Figure 2.5 Bitcoin block propagation
From Faria & Correia (2019)

Node A announces a new block to its neighbours by sending the NewBlockHashes in Figure 2.6. It is Node B that responds to this by sending the block header to Node A through a message called GetBlockHeaders. To add a block to the chain, Node B validates the block body. Block announcements have more overhead than transaction announcements. Transactions messages are broadcast by Node A to its neighbors when it receives a new transaction. Nodes add new transactions to the queue whenever they receive them (Faria & Correia, 2019).

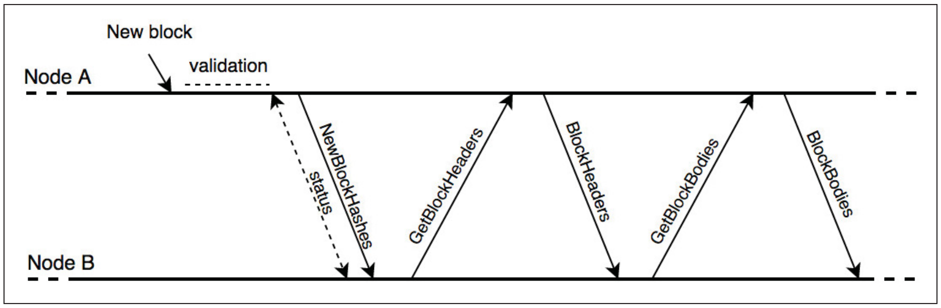


Figure 2.6 Ethereum block propagation FromFaria & Correia (2019)

2.2 Features of blockchain

The purpose of this section is to review the main features of blockchain that are relevant to our study and specifically associated with fork probability.

2.2.1 Fork

Forks can be divided into two main categories: intentional forks and accidental forks. Intentional forks are typically undertaken in order to resolve technical problems, recover lost resources, and add new features (Chang *et al.*, 2019).

An accidental fork occurs when blocks at the same height are mined almost simultaneously, resulting in the generation of a different block before the overall block propagates to the entire network (Haque & Rahman, 2020). The network is therefore extended with two distinct chains. In the event of a fork, each node has a different block from the latest block, resulting in inconsistent data. When the difficulty of block generation is increased and the interval between block generation is increased, the probability of generating simultaneous blocks is decreased. In turn, this reduces the likelihood of forks occurring. The Bitcoin blockchain generates a block on average every 10 minutes, depending on the difficulty of generating blocks.

As a result of strictly designed protocols that are difficult to change, many intentional forks have already occurred. This makes it impossible for blockchains to come to a consensus in case of conflict. (Seike, Aoki & Koshizuka, 2019). The first step in blockchain design is to specify the parameters and properties of the protocol, which are undoubtedly relevant to the community and future approaches of users. In spite of this, it is not possible to modify such a protocol in order to provide users with an easily upgradeable blockchain. Because of the promising results of blockchain technology, such a modification is not possible or even necessary. In the event of conflicts or upgrades, it is challenging for most of them to change protocols and reach a consensus. As a result, forks result in the scattering of the community and users.

As soon as a blockchain is established, any changes are tricky and complicated. This is regardless of whether they are made in the transaction layer or the protocol layer, although changes are crucial in the event of bugs or upgrades. When changes need to be made to a blockchain protocol, a fork is almost certain to occur. As a result of a fork, a blockchain diverges towards two potential paths in terms of the history of transactions or the rules used to determine whether a transaction is valid (Chason, 2019). Following the application of the revised rules to the blockchain, the chain will split into two separate chains, the original chain and the forked chain.

There are two types of intentional forks in blockchain technology, namely hard forks and soft forks. During a hard fork, the old rules are no longer available, but during a soft fork, the new rules are still valid for the miners. Unlike a hard fork, a soft fork involves a change of rules that still results in updated blocks that are recognized by the old software. In spite of this, the revised rule is not compatible with the older software, and some members of the community decide to adhere to the old rules. In the mining process, there is a case called a temporary fork, which is unimportant to us. In this case, the puzzle will be solved simultaneously by two miners. Temporary forks of this type will automatically disappear over time.

Bitcoin has experienced many intentional forks since its establishment as the first cryptocurrency and blockchain protocol. Approximately 74 of the 105 fork projects are active at this time (Kwon, Kim, Son, Vasserman & Kim, 2017). Additionally, altcoins, which are cryptocurrencies that are similar to Bitcoin, offer the same type of experience.

Due to network delays and traffic, accidental forking is one of the most challenging challenges in a blockchain system. In some cases, blocks are mined at a conflicting time, causing two or more branches to grow simultaneously. This refers to the construction of a new branch from a specific point in the main chain. The network will resolve the conflict by selecting the longest chain and discarding the blocks of the other chains as orphan blocks. Through analysis of different influencing parameters and modeling of the fork occurrence probability, this study attempts to reduce the number of accidental forks. As a result, we will refer to a fork as an accidental one from this point forward.

It is all about loss when a fork occurs in a blockchain-based system. A potential security issue may result from interfering with the concept of consistency and immutability of the blockchain network. As a result, a cryptocurrency may suffer a significant loss of confidence, lose its exchange rate, reduce trust in the network, waste a significant amount of computing resources, and be prone to attacks, double-spending, and fake blocks.

As a result of the propagation delay, each node has its own replica of the blockchain, which may differ from the replicas of other nodes. There may be forks in the blockchain ledger if multiple nodes have different views of the ledger. Consensus protocols are responsible for resolving forks in blockchain-based systems. In order to determine which blockchain (fork) should be the global chain, different consensus protocols utilize different rules. When a fork occurs in Bitcoin, the longest chain is selected to merge the fork. Based on the local blockchain for all nodes, we select the longest (deepest) blockchain to model this protocol. We resolve forks when multiple nodes have different blockchains with the same depth through a voting mechanism in which nodes vote on the longest fork. In this regard, the fork with the most votes is considered to be the longest, and thus, the global chain is considered to be the longest.

2.2.2 Network delay

The network delay (D) parameter, introduced by (Shahsavari *et al.*, 2019a), is the sum of different delays in the network including:

1. *t_v*: verification time.
2. *t_inv*: the arrival time of the *inv* message in destination node.
3. *t_get_header*: a message to the origin node asking for the newly mined block header, *get_header*.
4. *t_header*: a message *header* is sent by the origin node to the neighbor node through which the new block header arrives.
5. *t_get_block_body*: the arrival time of the message *get_block_body* through which the receiver node asks the sender to send the block body.
6. *t_block_body*: the arrival time of the message *block_body* through which the origin node sends the body of the new block to the neighbor node.

The formula of *D* is shown in Equation 2.1, with details in Figure 2.7.

$$\mathbb{D} = t\_v + t\_v + t\_inv + t\_get\_header + t\_header + t\_get\_block\_body + t\_block\_body$$

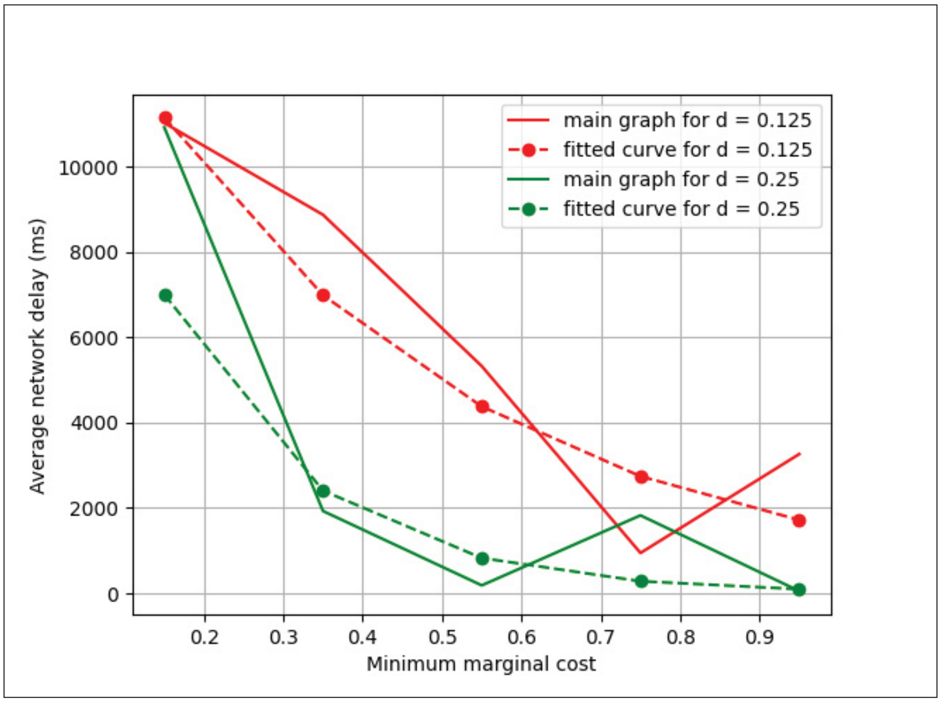


Figure 2.7 Network delay (*D*) calculation

2.2.3 51% attack

Blockchain-based cryptocurrency blocks are almost impossible to manipulate. In the event of a change (such as a fork or manipulation of data), it is used as a distributed consensus mechanism to determine the valid chain. The purpose of this mechanism is to determine whether the old chain or the newly created chain is the valid chain. Proof of Work (PoW) is one of the most common consensus mechanisms used by blockchains. An attacker who controls more than half of the total computing power in the P2P network can manipulate transactions in a malicious manner.

A 51% attack causes double-spending issues, selfish mining, etc. (Bae & Lim, 2018; Conti, Kumar, Lal & Ruj, 2018). The advantage of sharded networks is that they distribute computing power over multiple nodes, making them more susceptible to 51% attacks. Despite the fact that this issue is critical, and that several studies have been conducted to address it (Sirer & Eyal, 2014; Solat & Potop-Butucaru, 2016; Ruffing, Kate & Schröder, 2015), it is not the primary focus of this study.

2.3 Validation degree

The validity degree was introduced by the authors in the blockchain (Liu *et al.*, 2019a), which determines the percentage of nodes that verify a new block. In the example above, when v is equal to 0.5, half of the nodes validate the new received block. In contrast, the other half accepts the newly received block without validating it. When $v = 0.75$, only a quarter of nodes accept new blocks without validation. An increase in the validation degree will result in an increase in network delay as more nodes become involved in the process of verifying blocks. According to the following equation, the validation degree can be used to estimate the network's security assessment rate:

$$\rho = \tau \frac{r_s^2}{v} \tag{2.1}$$

It should be noted that τ represents the synchronization time and r_s represents the stale block rate, respectively. Time between the excavation of the first block and the acceptance of the last block is referred to as synchronization time.

Based on a joint analysis of the block's security and performance at different validation levels (Liu *et al.*, 2019a), the security assessment rate is determined. This parameter illustrates the trade-off between the probability of a fork occurring and the level of security. In the case of a decrease in validation degree, fewer nodes are required to verify a block. Consequently, if a node accepts a fake block without verifying it, that block will remain in the local chain until another block verifies and rejects it. Therefore, the nodes that have accepted the fake block should remove it from their local chains. Due to this, the network may be exposed to security risks. In spite of the benefits of reducing the validation degree with regard to the fork problem, it can pose a threat to the network as a whole. Each of our experiments has measured the security assessment ratio.

2.4 Theta

By using this method, it is possible to determine how many nodes are required to continuously validate newly created blocks (Liu *et al.*, 2019a). In order to understand how θ works, one must be aware that it represents the possibility of a block being untrusted by up to θ nodes. Upon accepting a block without verifying it, a certain number of consecutive nodes must verify it.

2.5 Fork probability models

In (Shahsavari *et al.*, 2019b), they examine the probability of a fork occurring based on the block propagation delay and the block generation interval. An illustration of their model can be found in equation 2.2.

$$\mathbb{F}(t) = P(T \leq t\_prop) = \int_0^{t\_prop} f(t) * d_t = 1 - e^{(\frac{t\_prop}{t\_B})} \tag{2.2}$$

According to the following equation, F(t) is the probability of forking, *t_prop* is the time required for a block to propagate over the network, and *t_B* is the time required to mine a new block. It is possible to model a series of discrete events whose average timing is known but whose exact timing is unknown by using this formula. A correlation cannot be established between the arrival time of an event and the arrival time of the previous event. Generally, it follows an exponential distribution. Based on these definitions, Equations 2.3 and 2.4 can be used to calculate the probability of a first event after an exact time and the probability of a first event during a specific period of time:

$$\mathbb{P}(T > t) = e^{-(\frac{events}{time})*timeperiod} = e^{-\lambda} \tag{2.3}$$

$$\mathbb{P}(T \leq t) = 1 - e^{-(\frac{events}{averagetime})*timeperiod} \tag{2.4}$$

Where we set the following values:

- *event* = 1
- *averagetime* = *t_B*
- *timeperiod* = *t_prop*

The following equation illustrates the corresponding fork evaluation model:

$$r\_s\_1 = 1 - e^{(-\frac{t\_prop}{t\_B})} \tag{2.5}$$

Due to the fact that block propagation time depends on several parameters, such as block size, bandwidth, average number of connections per node, and number of participating nodes, it would appear that this model cannot be used to accurately forecast forks. As a result, whenever the fork rate is mentioned in an experiment, it indicates that the experiment was conducted within a specific period of time during which the experiment was conducted.

2.6 Blocksim

In our work, we have extended the previously mentioned simulator Blocksim. This process consists of three steps: *Transaction*, *Block* and *Fork resolution*, as shown in 2.8 and 2.9. It is important to note that we have omitted the details of Blocksim, which can be found at (Faria & Correia, 2019).

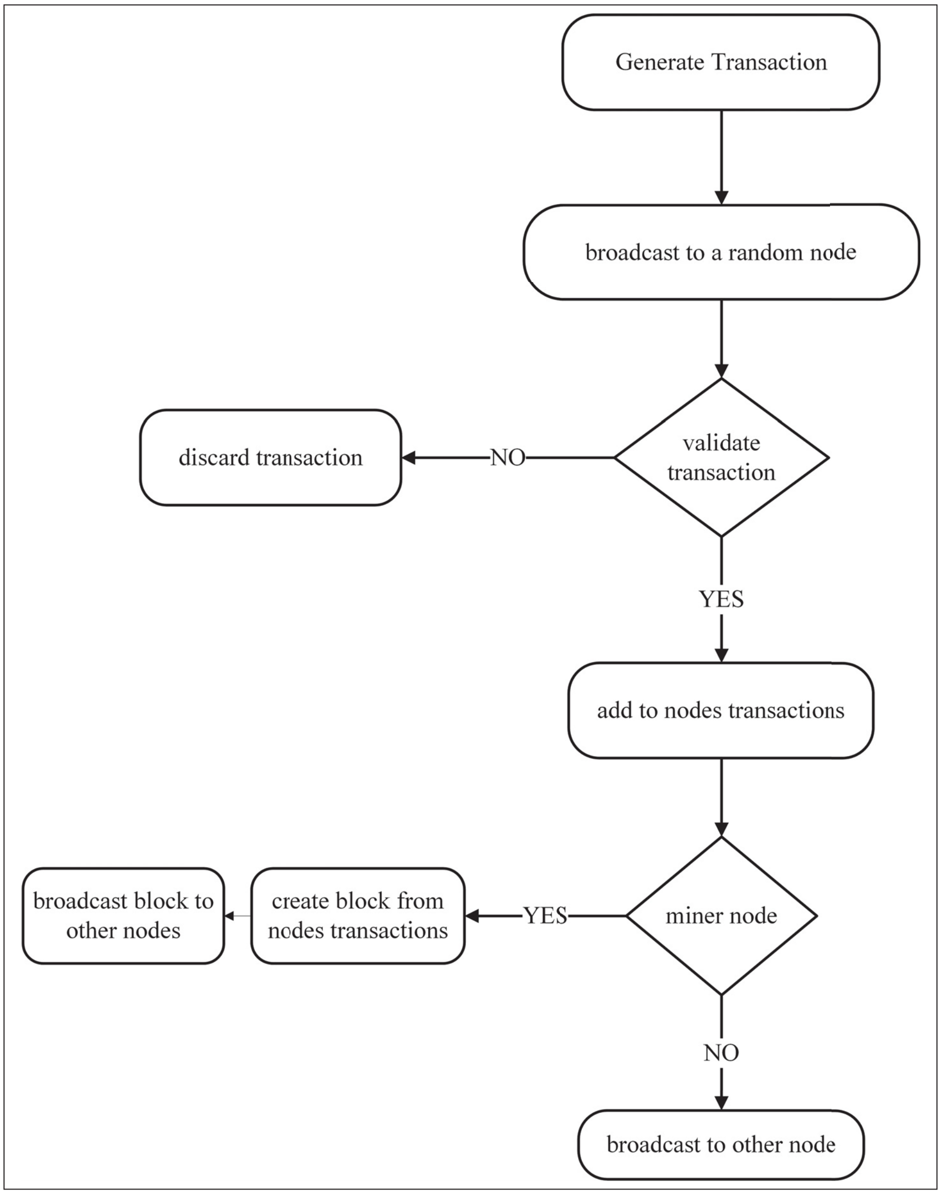

Generate Transaction
broadcast to a random node
validate transaction
NO
discard transaction
YES
add to nodes transactions
miner node
YES
create block from nodes transactions
broadcast block to other nodes
NO
broadcast to other node

Figure 2.8 Blocksim flowcharts for new transactions

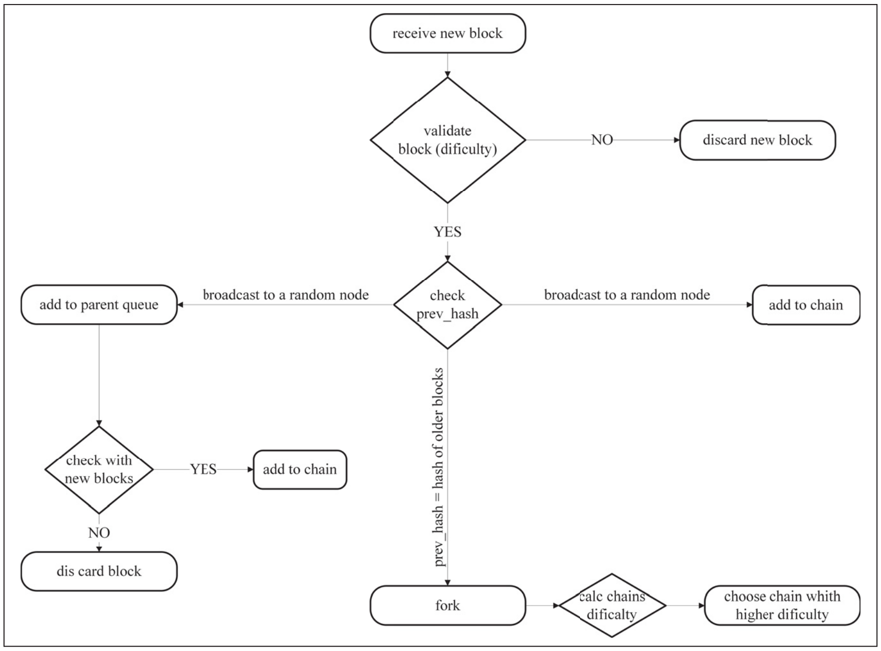


Figure 2.9 Blocksim flowcharts for new blocks

2.7 EIP-1559

Transaction fees are a component of the Ethereum protocol that determine the price paid by its creators for each transaction added to the blockchain. A first-price auction has been the policy of Ethereum since its inception for charging transaction fees: Each transaction is accompanied by a bid equal to the gas limit times the gas price, which is transferred by its creator to the miner of the block containing it (Roughgarden, 2020). The EIP-1559 proposes a significant change to Ethereum's transaction fee mechanism. A base fee serves as a reserve price, balancing supply and demand as part of the design. A block's base fee is paid by transactions included in the block; this fee is burned rather than transferred to the block's miner. A block size of 25M gas is the maximum; for example, with a target of 12.5M gas, there is no limit to the size of a block. Block fees are adjusted after each block, with larger blocks increasing them and smaller blocks decreasing them.

Users may request special treatment, such as immediate inclusion in a period of rapidly increasing demand or a specific position within a block. Transaction tips may be added to the base fee and are transferred directly to the miner of the block in which they are included (Roughgarden, 2020). It will be necessary for a user to pay two parts in order for a transaction to be included in a block. First, there is a fee cap that will be compared to the block's base fee.

$$\mathit{fee\_cap} \geq \mathit{base\_fee} \tag{2.6}$$

and there is also a premium which will be compared to the miner's marginal cost for the block to be accepted by the miner:

$$\mathit{premium} \geq \mathit{marginal\ cost\ of\ a\ miner} \tag{2.7}$$

The overall condition for each transaction is as follows:

$$\mathit{min}\ \{\mathit{fee\_cap}\ \text{-}\ \mathit{base\_fee}\}\ \geq \mathit{marginal\ cost\ of\ a\ miner} \tag{2.8}$$

To conclude, the miner of each block will receive the following tip:

$$\mathit{miner's\ tip} =\ \mathit{min}\ \{\mathit{fee\_cap}\ \text{-}\ \mathit{base\_fee},\ \mathit{premium}\} \tag{2.9}$$

Hence, the proposed reform to EIP-1559 presents a model for dynamically adjusting the base fee:

$$b_{t+1} = b_t * (1 + d * \frac{block\_size - target\_load}{target\_load}) \tag{2.10}$$

The base fee of the previous block is stated in b_t, the block size in gas is stated in $block/_size$, the step size is stated in d, and the last parameter is the old maximum size that can be doubled to $block\_size$. Typically, this occurs when the latest $block/_size$ value is smaller than the latest $target/_load$ value. Thus, each block determines the protocol state for the following block. However, there is an adjustment factor, or step size parameter (d), which is equal to 12.5%, which limits the increment and decrement of the base fee. The base fee will increase and decrease by approximately 12.5% after the maximum block size (twice the target size) and the minimum block size (an empty block).

Therefore, the purpose of EIP-1559 reform is to enhance users' experience of suggesting fees, alter the incentive nature of the fee market, and prevent fees from fluctuating due to network congestion. There are, however, some concerns raised by EIP-1559 that are worth mentioning. There was a competition over a single fee prior to EIP-1559. Meanwhile, in EIP-1559, with two fees, the base fee, the first one, is dynamically adjusted by the protocol, so there is no issue from this perspective.

However, we remain pessimistic about the possibility of achieving the goal of EIP-1559 based on the second part, which covers the marginal costs of the miner. As EIP-1559 only shifts the incentive mechanism from a single fee to the second component of EIP-1559's fee, the tip, it is unlikely to change the competitive nature of the fee market.

2.8 Sharding solutions

Sharding provides scalability by dividing the network into smaller subgroups. The number of shards increases as a result, reducing storage and communication requirements. Sharding models can be categorized into two main categories: transaction sharding and state sharding (Yun *et al.*, 2019).

2.8.1 Transaction sharding

A blockchain system's scalability issue is addressed by transaction sharding. Using sharding technology, transactions are processed in parallel by nodes in shards, thereby enhancing the transaction throughput (Zhang, Wang, Li & Zhou, 2020).

1. *Elastico (Luu* et al.*, 2016):* It is one of the earliest blockchain sharding models and offers a new method of increasing the scalability of a blockchain. Through the random distribution of transactions among shards, parallel computing can be achieved. However, the model

requires nodes to store all transaction data, which poses a significant challenge to their storage capacities. Furthermore, it does not support cross-shard transactions.

2. *Zilliga (Aiyar, Halgamuge & Mohammad, 2021):* It is possible to process transactions in parallel by sharding a blockchain system, which improves the system's performance. The transaction arrival rate will be increased to a thousand times that of Ethereum as a result. A large number of cross-shard transactions also adversely affects the system's performance.

2.8.2 State sharding

The purpose of state sharding is to extend and improve transaction sharding. As a result of their scalability, blockchain systems are capable of scaling both in terms of computing power and storage capacity. Instead of storing the state data for the entire lockchain, each node stores the state data for its own shard. As a matter of fact, there are several state sharding models, including OmniLedger (Kokoris-Kogias *et al.*, 2018), RapidChain (Zamani *et al.*, 2018), Monoxide (Wang & Wang, 2019), and Chainspace (Al-Bassam, Sonnino, Bano, Hrycyszyn & Danezis, 2017).

1. *OmniLedger (Kokoris-Kogias* et al.*, 2018):* To reduce the number of cross-shard transactions, an atomic submission protocol is proposed. It is possible to reduce the storage requirements for the nodes within a shard by placing a state block within the shard. Nevertheless, it suffers from a long transaction delay when dealing with cross-shard transactions. Due to this, blockchain systems are impacted substantially in terms of parallelization.
2. *RapidChain (Zamani* et al.*, 2018):* It is an improvement over OmniLedger (Kokoris-Kogias *et al.*, 2018). By using the intera-shard routing protocol, cross-shard transactions can be processed more quickly. It is possible to significantly increase the scalability of a blockchain system by utilizing RapidChain. When there are a large number of false cross-shard transactions, the verification of transactions may be delayed in a blockchain system.

2.9 Particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm has been widely used for the solution of single-optimization problems (Wen-Bin & Yong-Hong, 2021). The algorithm uses a random search method based on the behavioral pattern of a swarm of birds to forage together and flock together. Thus, it is affected by the individual and social behavior of the birds (i.e., particles) within a flock. This algorithm consists of the following stages: initialization, search, updating, and convergence of the best positions and values. There are several iterations required for each of these processes. The first step in solving an optimization problem is to generate a set of random particles or solutions (Nartey *et al.*, 2022).

Multi-objective optimization is the process of obtaining an optimal vector of variables within a feasible area that is defined by a set of constraints. A vector of variables is identified in such a way as to minimize or maximize a vector of objective functions (Nartey *et al.*, 2022).

Researchers have been using evolutionary algorithms to solve multi-objective problems, which are referred to as Multi-Objective Evolutionary Algorithms (MOEA) (Yang, Liu & Tan, 2021). A high degree of efficiency, robustness, and versatility can be achieved when these algorithms are applied to complex optimization scenarios. According to (Ridha *et al.*, 2021), multi-objective optimization was used to solve the problem of designing a standalone photovoltaic system (Nartey *et al.*, 2022).

As a solution to multi-objective optimization problems, Coello and Lechuga introduced MOPSO in (Coello & Lechuga, 2002). Before its introduction, PSO was only capable of handling single-objective optimization problems. Multi-objective optimization problems were reduced to single-objective problems using a weighted sum approach (Zhang, Sheng, Lu & Shen, 2021b).

MOPSO keeps track of non-dominated solutions as well as updating the population during each iteration. To arrive at a Pareto optimal set, these elements are referred to as the Pareto set. In order to create a complete archive, elements are added until the archive is complete. Once the archive has been sorted, only the very best solutions are retained. There would be more than one optimal solution for the global best search result due to the fact that there are multiple objective functions. The result is the selection of a global leader, which is used to adjust all other solutions (Nartey *et al.*, 2022). The flow chart for the MOPSO algorithm is illustrated in 2.10.

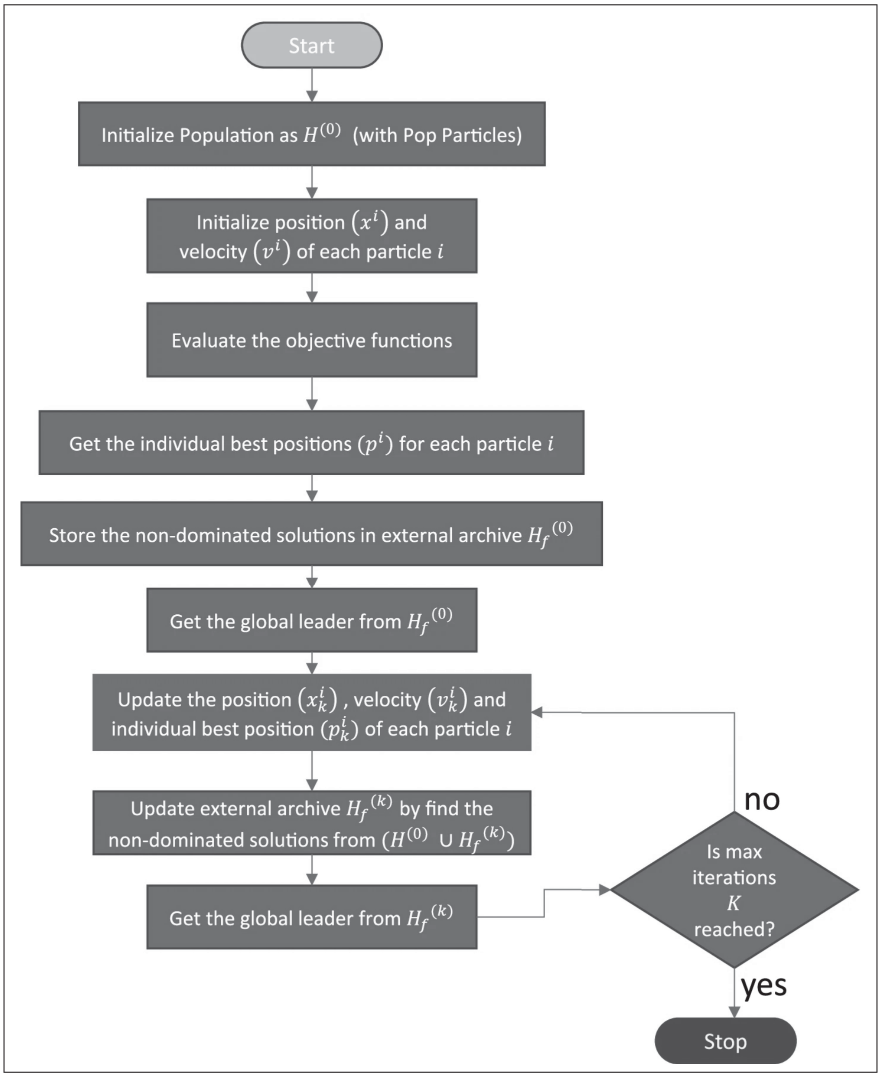

Start
Initialize Population as $H^{(0)}$ (with Pop Particles)
Initialize position (x^i) and velocity (v^i) of each particle i
Evaluate the objective functions
Get the individual best positions (p^i) for each particle i
Store the non-dominated solutions in external archive ${H_f}^{(0)}$
Get the global leader from ${H_f}^{(0)}$
Update the position (x_k^i) , velocity (v_k^i) and individual best position (p_k^i) of each particle i
Update external archive ${H_f}^{(k)}$ by find the non-dominated solutions from $(H^{(0)} \cup {H_f}^{(k)})$
Get the global leader from ${H_f}^{(k)}$
Is max iterations K reached?
no
yes
Stop

Figure 2.10 Flow chart for MOPSO
From Nartey *et al.* (2022)

The Particle Swarm Optimization (PSO) algorithm is based on mathematical modeling of species interactions within groups, such as those observed in bird flocks (Kennedy & Eberhart, 1995). This optimization algorithm has been used frequently in different fields to solve challenging and complex optimization problems using swarm intelligence (Thakkar & Chaudhari, 2021; Pervaiz, Ul-Qayyum, Bangyal, Gao & Ahmad, 2021; Haouari & Mhiri, 2021).

Using a population of individuals called particles, this algorithm searches for solutions within the solution space. Each individual or particle has the potential to provide a solution to an objective function. The particles identify the optimal solution through cooperative exploration and exploitation of different areas of the solution space. There are three parts to a particle: a position vector (the likely solution), a velocity vector indicating the direction in which the particle will move, and a memory that saves the particle's optimized position from the beginning to the current iteration. An illustration of particle movement within the solution space is provided by the following equation:

$$v(t+1) = w.v(t) + c_1.rand.(p\_best - x(t)) + c_2.rand(p\_gbest - x(t))$$

$$x(t+1) = x(t) + v(t) \tag{2.11}$$

In these equations:

- *v*: The velocity of the *i*th particle
- *x*: The position of the particle
- *t*: The number of iteration
- *c*: Learning factors designed to control the exploring and exploiting capability of PSO
- *rand*: A positive random number between 0 and 1 under normal distribution
- *w*: The inertia weight coefficient
- *p_best*: The best position of the particle from the beginning to current iteration
- *p_gbest*: The position of the best particle in each iteration which has the best fitness value in the population.

The particles are initially initialized randomly in the search space at the beginning of the process. A cost function is then used to calculate the fitness of each particle, and the particle with the highest fitness value is selected as the leader of the population (*p_gbest*). The following steps are followed in each iteration until the termination criteria are met:

1. Equations for moving particles in space 2.11 and 2.11.
2. Using the cost function to calculate the fitness value of each particle.
3. Update of the leader's position and the particle's memory (*p_best*).

The solution to the optimization problem will be presented in the final step as the particle (or leader) with the highest probability in the last iteration. Particle swarm optimization is illustrated by the following pseudocode.

Algorithm 2.1 Pseduo code for the PSO algorithm

```
Initialize pop
for $iteration \leftarrow 1\ to\ itmax$ do
    for $particle \leftarrow 1\ to\ popmax$ do
        $velocity \leftarrow$
          $w * velocity + cl * rand * (p\_bestparticle) + c2 * rand * (p\_gbestparticle)$
        $particle \leftarrow particle + velocity$
        $fitness \leftarrow costfunction(particle)$
        if $fitness_{particle} > costfunction(p\_gbest)$ then
            $p\_gbest \leftarrow particle$
        end if
        if $fitness_{particle} > costfunction(p\_best)$ then
            $p\_best \leftarrow particle$
        end if
    end for
end for
$final\_solution \leftarrow p\_gbest$
```

Particles tend to move toward the most suitable particle in the population (p_gbest) when c_2 is high and c_1 is low, as described by equations 2.11 and 2.11. The particles will search around the most optimal position that they observed at the beginning of the iteration (p_best) if c_1 and c_2 are high. In order to control the algorithm's exploratory and exploitative capabilities, it is necessary to select the appropriate amounts for these two factors. According to different publications, fixing them at 2 will result in a reasonable convergence rate.

It is true that the PSO algorithm has the advantage of fast convergence (Wen-Bin & Yong-Hong, 2021), however, it has a serious disadvantage in that it tends to get trapped in local optima. This often leads to an excessive amount of searching.

Also, a parallel PSO algorithm reduces the computational costs associated with the algorithm. It is important to note that the majority of these parallel algorithms are not distributed, which means that they are composed of multiple processors communicating with each other through shared memory rather than multiple computers communicating through a network. Additionally, a great number of parallel implementations of the PSO algorithm are based on synchronous implementations, which evaluate all particles within each iteration before moving forward. It is possible for this implementation to result in poor parallel performance in some cases, particularly if the cluster's resources are imbalanced. Many works have proposed a distributed asynchronous PSO algorithm based on Message Passing Interface, but have failed to consider fault tolerance and scalability.

CHAPTER 3

THE FORK PROBABILITY MODELING OF BLOCKCHAINS

The content of this chapter concerns a novel form modeling that incorporates parameters not previously considered, such as network delay and validation degree. Section 3.1 presents various motivating use cases for this work. Section 3.2 explains the proposed model for Ethereum network and EIP-1559. Section 3.3 presents the proposed model evaluation. Finally, a conclusion is offered in Section 3.4.

3.1 Introduction

The focus of this research is on forks, one of the most significant challenges facing blockchain networks. Intentional forks and accidental forks are the two main types of forks. Through these intentional forks, technical issues have been resolved, resources have been recovered, and updated features have been added (Chang *et al.*, 2019).

As a result of propagation delays, each node has a different replica of the blockchain. Forks are resolved by blockchain-based consensus protocols. Different consensus protocols are used to determine the global chain. Bitcoin PoW resolves forks by choosing the longest chain. According to the paper, the longest (deepest) blockchain is selected for all nodes in order to model this protocol. Using a voting mechanism, multiple nodes with the same depth blockchain can fork. There is a depth to the fork that is determined by the nodes. Therefore, the longest fork is considered to be the global chain.

The newly developed fork rate model enabled us to estimate the probability of forks. Furthermore, we were able to determine whether fork rates increased or decreased. The proposed probability model has been used to analyze the effects of EIP-1559's critical parameters on fork probability. A fee market mechanism was introduced in EIP-1559 in 2021 to improve the user's experience in setting fees for transactions (Reijsbergen *et al.*, 2021).

This research has made the following contributions:

- A new blockchain simulator by adding new capabilities, such as the ability to simulate *validation degree* and the ability to simulate EIP-1559 networks.
- It is being investigated how new and important parameters affect the fork issue.
- This new model includes more parameters that affect fork occurrence probability than previous models.
- Analysis of the impact of EIP-1559 parameters on the probability of occurrence of forks based on the proposed model.

3.2 Fork probability modeling

We have developed our fork occurrence probability model on the basis of the model described in the previous section. Although the previous method can estimate the likelihood of forking, its main disadvantage is that it excludes other critical parameters, such as network delay. A number of parameters have a significant impact on the probability of a fork occurrence, including validation degree (v), network delay (D), time interval between blocks (*block_interval*), and θ.

Delay on the network is affected by the level of validation. Increasing the validity degree leads to a higher number of nodes participating in the block validation process, resulting in a longer network delay. On the other hand, an increase in network delay increases the probability of multiple nodes accepting the same block at the same time. There is an increasedelihood of forks occurring when the network is delayed. In a similar manner to θ, *block_interval* has an opposite effect on the probability of a fork. Based on this theory and the previously introduced models, we have developed our fork model as follows:

$$r\_s\_3 = 1 - e^{\left(-\frac{D*v}{t\_B*block\_interval*\theta}\right)} \tag{3.1}$$

We calculate the forking rate of blocks in accordance with Equation 3.1, in which we account for the delay in the network (D), the degree of validation (v), and the average time interval between blocks (t_B).

In comparison with previous models, our proposal has the advantage of incorporating more effective parameters, such as the network delay. Forks are primarily caused by network delays, which can be affected by a variety of factors. As previously mentioned, the validation degree is positively correlated with the network delay. The parameters contained in EIP-1559 may also influence the network delay and, consequently, the likelihood of forking.

As a result of a reduction in miners' marginal costs, block creation time is likely to increase. A fixed period of time will result in fewer blocks being created and propagated over the network. In order to reduce network delays, it is possible to increase the average marginal cost of the miners, which results in a lower likelihood of forks.

For the purpose of evaluating the performance of the proposed model, we developed a new blockchain simulator by adding new input parameters, including the validation degree and θ. Our newly developed feature allows users to analyze the impact of these key parameters on various aspects of the blockchain. To determine whether an increase in validation degree will lead to an increase in the number of forks, we can simulate the network, for example. Additionally, we are able to measure and analyze the security assessment rate of the blockchain in various situations by adding the validation degree to the simulator.

As part of our enhancements, we have included the capability to simulate the EIP-1559 network. A fee cap is assigned to each transaction in the updated simulator based on a random process over an interval defined by the user. In contrast, each miner may select the marginal cost and block size at random within predetermined intervals. For simulating the block fee, Equation 2.10 is used to calculate the base fee, and a transaction may be added if it meets the transaction condition in Equation 2.8. This allows our solution to simulate the EIP-1559 network with a variety of different parameters in order to determine how they affect the network. In order to make it applicable for beacon chain blocks as well, some modifications must be made to the new simulator. It is important to note that the new simulator is still compatible with POW-based Ethereum networks.

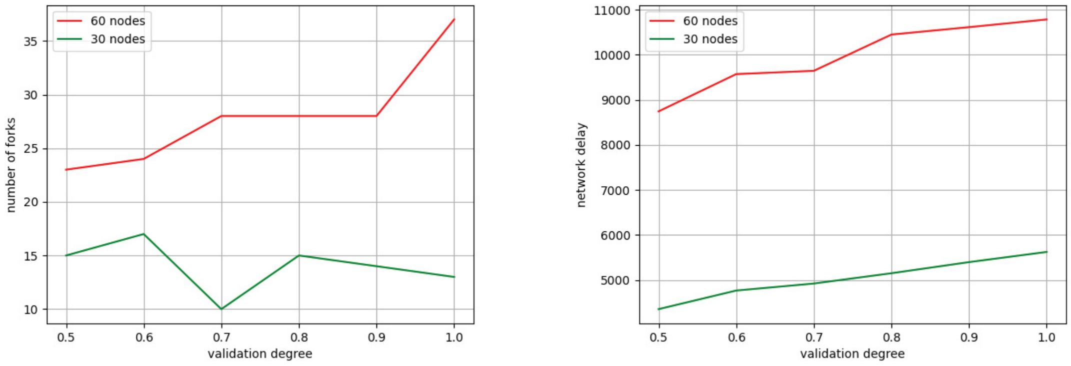
Using our improved simulator, we investigated the impact of changing miners' marginal costs and step sizes (see Equation 2.10) on the network delay. This research aims to introduce a new version of blockchain simulator that includes more network parameters and can also simulate the EIP-1559 network. Therefore, this simulator offers the following advantages:

1. Provide users with more input parameters, such as validation degree, θ, minimum marginal cost of a miner, fee cap, step size, etc.
2. The proposed fork probability function will be used to calculate the fork occurrence probability and the security assessment rate during the simulation.

In order to determine the impact of different parameters on the probability of a fork occurring in the Ethereum network and the EIP-1559 network, we conducted several experiments using the enhanced blockchain simulator.

3.3 Evaluation and results

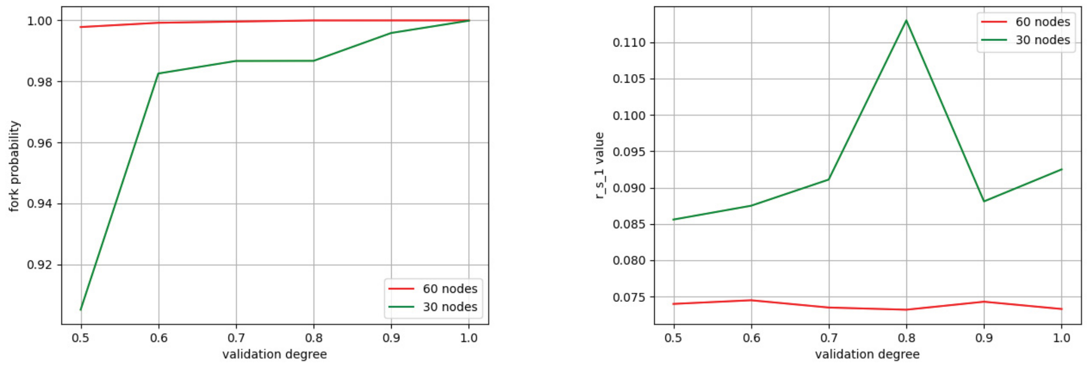
A fork occurrence analysis has been conducted on both the normal Ethereum network and the EIP-1559 network. The results achieved will be discussed in more detail in the following subsections.



a) 30 and 60 Nodes average forks

b) Effect of validation degree on *D*

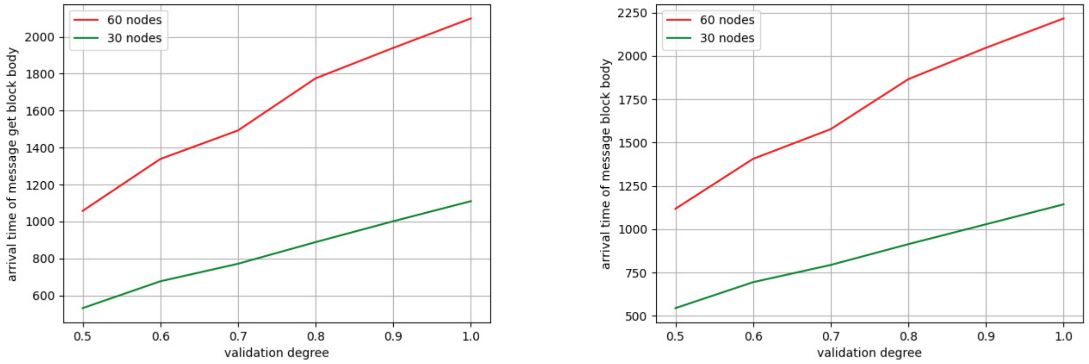
Figure 3.1 Fork and delay charts



a) 30 and 60 node validation degree impact on r_s_3

b) 30 and 60 node validation degree impact on block *r_s_1*

Figure 3.2 The rs1 and rs3 chart



a) 30 and 60 node validation degree impact *t_get_block_body*

b) 30 and 60 node validation degree impact on *t_block_body*

Figure 3.3 Block body comparison

3.3.1 An analysis of the results of a regular ethereum network

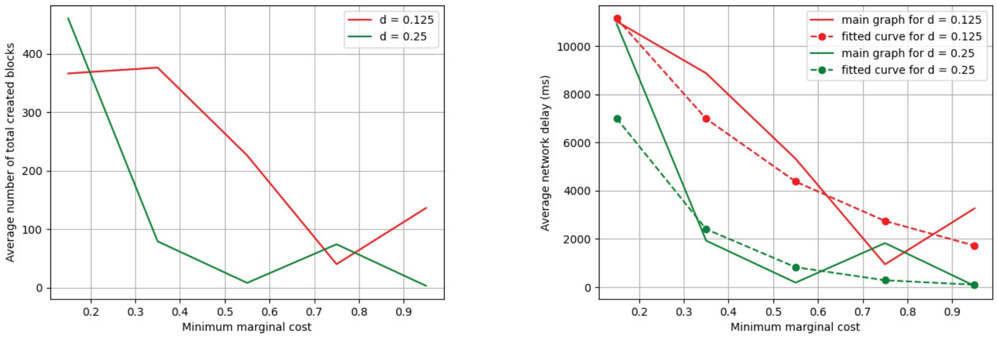
It was our intention to evaluate our model by conducting two experiments. First we had 30 nodes in the first experiment, then we had 60 nodes in the second experiment, and finally we had 120 nodes in both experiments. In both cases, there were a large number of mining nodes that were connected. The average results obtained during the two experiments can be found in Tables 3.1 and 3.2.

Table 3.1 An average of D, ρ, r_s_1 and r_s_3 for 30 nodes

V	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
D	4351.92	4764.59	4920.012	5148.73	5395.78	5621.69
ρ	1.94	2.088	1.56	1.18	1.37	1.29
r_s_1	0.0856	0.0875	0.0911	0.113	0.0881	0.0925
r_s_3	0.9051	0.9825	0.9866	0.9867	0.9958	0.9998

Table 3.2 An average of D, ρ, r_s_1 and r_s_3 for 60 nodes

V	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
D	8742.068	9569.83	9642.723	10447.41	10611.95	10783.072
ρ	4.126	3.79	3.25	3.107	2.84	2.5
r_s_1	0.074	0.0745	0.0735	0.0732	0.0743	0.0733
r_s_3	0.9978	0.9992	0.9996	0.9999	0.9999	0.9999

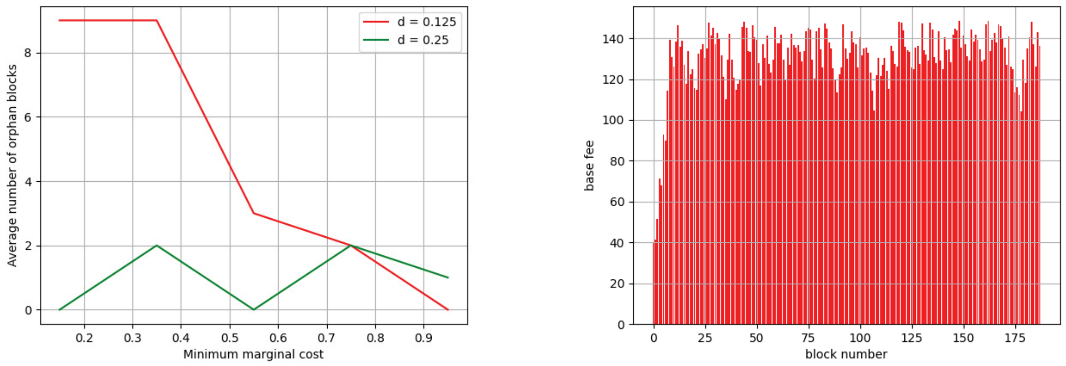


a) Average total blocks created at different marginal costs

b) Different marginal cost values and average network delay

Figure 3.4 Network delay and marginal cost

As shown in these tables, the validation degree affects the network delay (*D*), the security assessment rate (ρ), *r_s_1* and our fork probability model (*r_s_3*. For a network with 30 nodes, if v = 0.5, the probability of forking is 90.51%, whereas if v = 1, it is 99.98 %. For a network containing 60 nodes, these two values are estimated to be 99.78% and 99.99%.



a) Number of orphan blocks and marginal cost

b) Simulated base fee graph

Figure 3.5 Base fee and orphan blocks

The results of these tables suggest that an increase in validation degree will result in higher values of *D* and *r_s_3* and lower security assessment rates. As a result of increasing the validation degree, the number of nodes participating in the verification process increases significantly, resulting in a delay in the block propagation process. As a result, the blocks that were mined simultaneously have more time to propagate through the network and be accepted by other nodes as the head of the chain.

As shown in the tables 3.1 and 3.2, a higher validation degree raises the probability that forks will occur. The increased validation degree, however, decreases the likelihood that fake blocks will be accepted, as it increases the likelihood that nodes will verify the block. As can be seen from the values of the security assessment rate, this concept is clearly evident. The purpose of this parameter is to control the trade-off between the probability of a fork occurring and the security of a network. In this study, one of the key contributions is the implementation of this effective parameter in our blockchain simulator.

The tables 3.1 and 3.2 demonstrate that the network with more nodes experiences a longer delay since more nodes are required to verify and validate newly created blocks. As a result, there is a higher fork probability as well as a higher rate of security assessments as the number of nodes increases. Thus, an increase in the number of nodes can negatively impact the network situation in terms of the probability of a fork occurrence and the acceptance of fake blocks. In general, our fork model (*r_s_3*) performs well when it comes to predicting fork probability in different scenarios, whereas *r_s_1* has not been able to accurately predict fork probability. Because our model includes more effective parameters than *r_s_1*, it is more effective than *r_s_1*.

A summary of the average number of forks for 30 and 60 nodes, as well as for six different values of validation degree, is presented in Figure 3.1a. It is evident from these graphs that the number of forks increases with an increase in the level of validation.

There is a comparison between the fork probability and the network delay in Figures 3.2a and 3.1b. A significant network delay results from an increase in the validation degree and the number of nodes. There is a greater chance of a fork occurring as a result.

In addition, Figures 3.3a and 3.3b provide similar information about *t_get_block_body* and *t_block_body*. There is a longer delay between sending and receiving *get_block_body* and *block_body*. In addition to the increase in validation degree, increasing the number of nodes participating in the verification process will also increase the number of nodes participating in the verification process. Thus, the block miner should send and receive the corresponding messages (*inv*, *get_header*, *header*, *block_body*, and *block_body*) to a greater number of nodes. Consequently, it takes longer for a block to propagate over the network. As a result, there is clearly a greater value to *D*. In a similar manner, Figure 3.2b illustrates how *r_s_1* evolves over time. Based on the Figure, it is evident that *r_s_1* has a relatively low accuracy in predicting the likelihood of a fork. All of the experiments have been run at 6 and 15 respectively for θ and *Block_interval*.

3.3.2 Simulation results and analysis for EIP-1559

We conducted several experiments to determine whether marginal costs and step sizes affect network delay. We increased the marginal cost from 0.1 to 0.9 in two cases: $d = 0.125$ and $d = 0.25$. In these experiments, it is important to note that:

- Each transaction has a random *premium* value between 0 and 1.

- There may be 10 to 100 transactions in each block, which are also randomly generated during simulation.
- A total of 60 nodes, including 30 miners, are used in all of the simulations.

As shown in Figure 3.4a, marginal cost affects the total number of blocks created. According to this figure, as the marginal cost of miners rises, the number of blocks that can be created will decrease as a result of a decrease in the number of miners. Due to this, marginal costs go up, and there will be a decreased likelihood of a block to include transactions as a result. In addition, it is also possible to do the same thing by increasing the step size in order to achieve a similar effect. If there is a larger step size, the base fee per block will be higher than if there is a smaller step size.

The model we proposed was validated by hours of actual statistics taken from the Ethereum network, in order to validate its accuracy. In order to demonstrate the effectiveness of our improved simulator, we analyzed the marginal costs of the blocks created on the 19th of May 2022 between 2:49 and 5:49 AM EST and on the 21st of May 2022 between 7 and 11 PM EST on the 19th of May 2022. This is done by selecting a random sample of periods at random for the purposes of this analysis. As shown in the tables 3.3 and 3.4, an increase in marginal cost may result in the creation of fewer blocks on average.

The table 3.4 indicates that 483 blocks were added to Ethereum's main chain during the time period from 9 p.m. to 11 p.m. on the 21st of May. There were 495 blocks created between the hours of 5 p.m. and 7 p.m. During these two periods, the marginal cost for miners was 5.83 and 5.59. There were fewer marginal costs between 5 p.m. and 7 p.m., resulting in 12 more blocks being generated. This concept is illustrated in our simulation. According to Figure 3.4a, higher marginal costs result in fewer blocks being created.

Network delays are affected by marginal costs and step sizes, as shown in Figure 3.4b. As can be seen from this Figure, higher marginal costs and larger steps result in a reduction in network delay as fewer blocks are created. In the event that marginal costs and base fees are increased, miners will require additional time to complete their blocks.

Table 3.3 Block marginal costs from 2:49 AM to 5:49 AM EST on 19th of May 2022

Time of creation	Blocks ID	Num created blocks	Avg marginal cost
4:49 AM-5:49 AM	14803024-14803326	290	5.3
3:49 AM-4:49 AM	14802758-14803023	260	5.67
2:49 AM-3:49 AM	14802502-14802757	250	7.72

Table 3.4 Block marginal costs on May 21st from 7 to 9 PM EST

Time of creation	Blocks ID	Number created blocks	Avg marginal cost
9 PM-11 PM	14819475-14819973	483	5.82
7 PM-9 PM	14818956-14819474	503	4.68
5 PM-7 PM	14818445-14818955	495	5.59

Therefore, fewer blocks are propagated over the network, resulting in a shorter delay. Figure 3.4b illustrates the relationship between the marginal cost of miners and network delay. Two exponential curves have been fitted to values of the network delay in order to determine the relationship between the marginal cost of miners and network delay. With d = 0.125 and d = 0.25, the MSE error is 2.016 and 3.675, respectively. According to these curves and the MSE error values, an increase in the miners' minimum marginal cost leads to a significant reduction in the network delay. As can be seen from the Figures 3.4a and 3.4b, these two parameters are closely related. Reducing the network delay can reduce the probability of a fork.ity. By increasing marginal costs, it is possible to reduce the probability of forks occurring.

In addition, we examined the impact of marginal cost and step size on the number of orphan blocks. Increasing the marginal cost and step size generally reduces the average number of orphan blocks. This indicates a lower likelihood of a fork, as shown in Figure 3.5a. According to the following Figure, the proposed fork model is highly efficient. Figure 3.5b illustrates the graph of the base fee during the simulation, which illustrates the accuracy of our simulation.

3.4 Summary

In this chapter, a new model of the probability of fork occurrence is presented that includes more effective parameters in comparison to other methods such as the network delay (D) or the validation degree (v). To verify the validity and efficiency of the proposed model, we conducted experiments on the Ethereum network as well as the EIP-1559 network. In our simulation results, we found that increasing the validation degree will result in a significant propagation delay, which will increase the likelihood of a fork. In addition, the results indicate that a decrease in the number of blocks created in EIP-1559 will result in a higher probability of a fork caused by an increase in the marginal costs of the miners.

CHAPTER 4

SHARDING AND ITS IMPACT ON FORK PROBABILITY

In this chapter, our primary goal in this study is to determine the impact of adding new shards to a blockchain on the probability of forks occurring. In order to achieve this goal, we first developed a novel simulator which enables us to simulate sharded networks. Section 4.1 presents sharding and scalability issues after EIP-1559 update on Ethereum network. Section 4.2 explains the proposed model for Ethereum network with sharding capability. Section 4.3 presents the proposed model structure. Finally, a conclusion is offered in Section 4.4.

4.1 Introduction

Blockchain technology has encountered a number of challenges, including forks and low transaction processing rates in comparison with other payment methods. Additionally, sharding ensures that the network will be scalable in addition to providing a solution to the issue of low processing rates. In spite of this, it is unclear whether it will affect the likelihood of a fork. This chapter aims to determine whether adding new shards to a blockchain increases the likelihood of forking. To accomplish this goal, we developed a novel simulator that simulates sharded networks. Following that, we examined the effect of sharding on the occurrence of forks. This research makes the following contributions:

1. Introduce a novel simulator for simulating the network of EIP-1559 with multiple shards
2. Investigate the impact of adding new shards on the probability of forking

4.2 Sharding mdesign and modeling

According to the previous section, this study is primarily concerned with investigating the impact of sharding on the probability of occurrence of forks. In order to accomplish this, we enhanced BlockSim, introduced by the author. (Faria & Correia, 2019), by adding the capability to simulate sharded networks. Additionally, this new simulator is capable of simulating the EIP-1559's network, which includes new features such as *base_fee*, *Miners' marginal cost*, and *fee_cap*. The *base_fee* of each block is calculated using the equation 2.10. Figure 4.1 illustrates the different components of BlockSim.

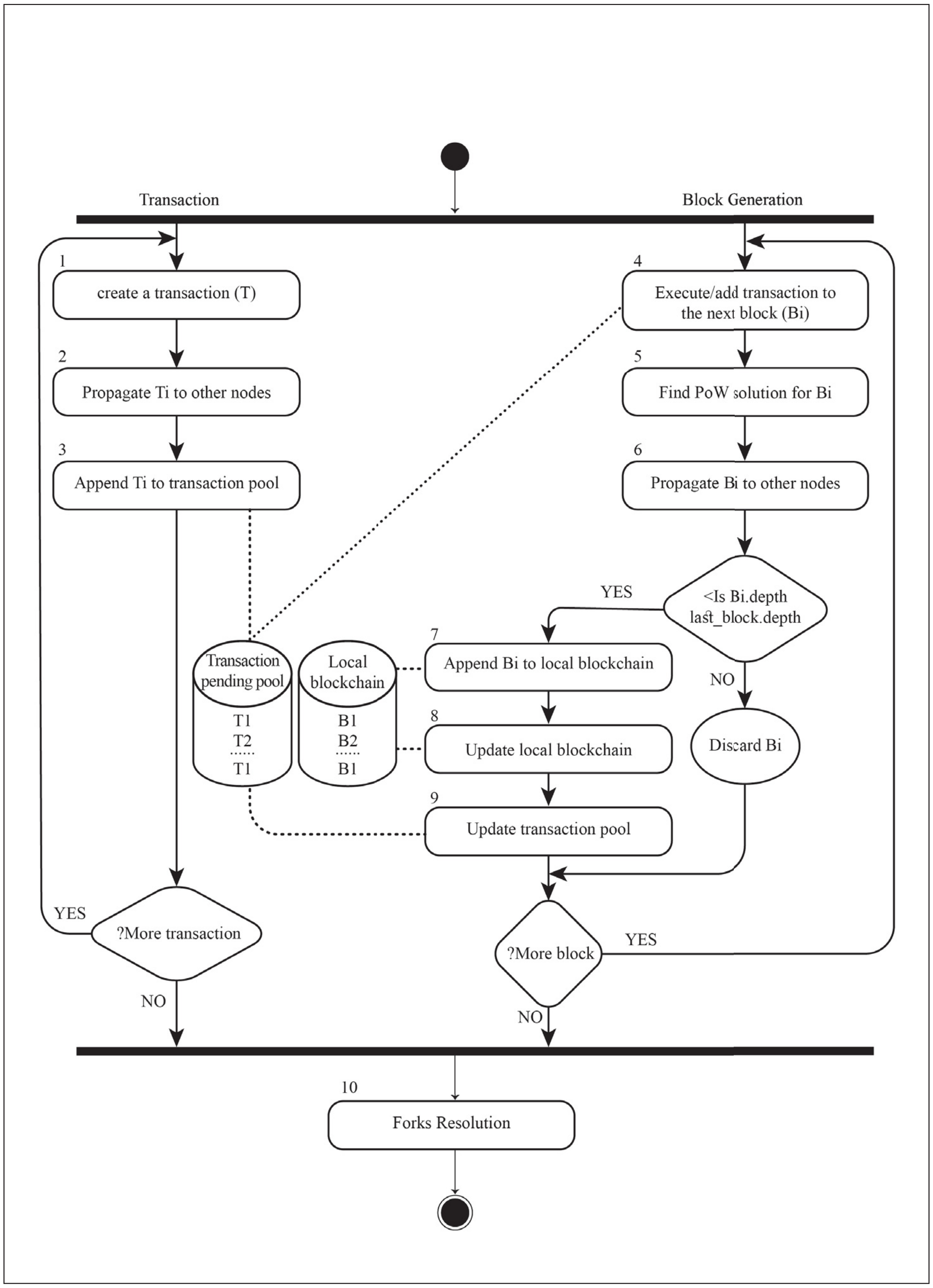


Figure 4.1 Different parts of the BlockSim
From Alharby & Van Moorsel (2019)

A new simulator has been developed that allows the user to determine the number of shards and then distribute nodes (miners and non-miners) uniformly among them. The process is then initiated independently by each shard. Every node has a new feature called *shard_num*, which identifies the shard to which it belongs. Additionally, there are two additional features in the transactions that determine the shard number for the sender and receiver.

For a sharded network to be effectively simulated, cross-shard transactions must be processed differently than in a non-sharded network. Transactions cannot be included in blocks until they have been confirmed both in the origin and destination shards. An additional feature called *confirmed* has been added to the transaction class in order to simulate cross-shard transactions. For intra-shard transactions, this feature is set to "true" so that each shard's miners can add intra-shard transactions to their blocks.

In cross-shard transactions, this feature is set to "false" at the beginning. When first-time miners pick up cross-shard transactions from their mempool, they do not append them to their blocks. As soon as the corresponding process is initiated, it adds the transaction to a repository in order to obtain confirmation from the origin and destination shards. A supplementary procedure has been added to the simulator in order to simulate cross-shard transactions. Before this procedure can be activated, a predetermined number of blocks must be created in each shard. If the corresponding parameter is set to 4, for example, the process is initiated after each of the four blocks has been created.

Through this process, cross-shard transactions are confirmed (their *confirmed* feature is set to "true") and returned to the miner's memory pool. By doing so, they will be able to be included in future blocks. The flowcharts of block creation and cross-shard transactions are displayed in Figures 4.2 and 4.3. Moreover, the proposed simulator presents a class diagram of Ethereum nodes 4.4.

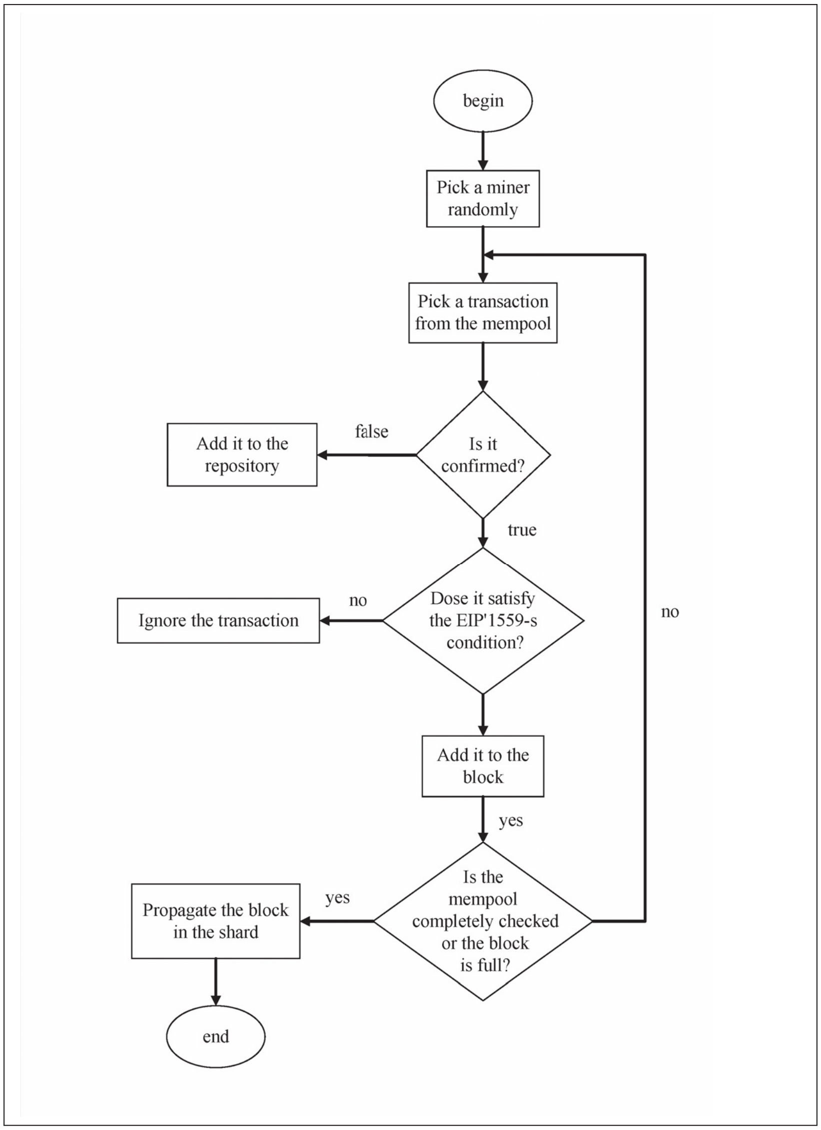

begin
Pick a miner randomly
Pick a transaction from the mempool
false
Add it to the repository
Is it confirmed?
true
no
Ignore the transaction
Dose it satisfy the EIP'1559-s condition?
no
Add it to the block
yes
yes
Propagate the block in the shard
Is the mempool completely checked or the block is full?
end

Figure 4.2 Flow chart of block creation in the new simulator

4.3 Evaluation and results

To determine the effect of sharding on the fork occurrence probability, a simulation of the EIP-1559 network with different shard numbers was conducted. The probability of a fork occurring (equation 3.1), the number of orphan blocks, and the overall delay of the network were calculated. In the following subsections, we present the results for networks with 60 and 120 nodes.

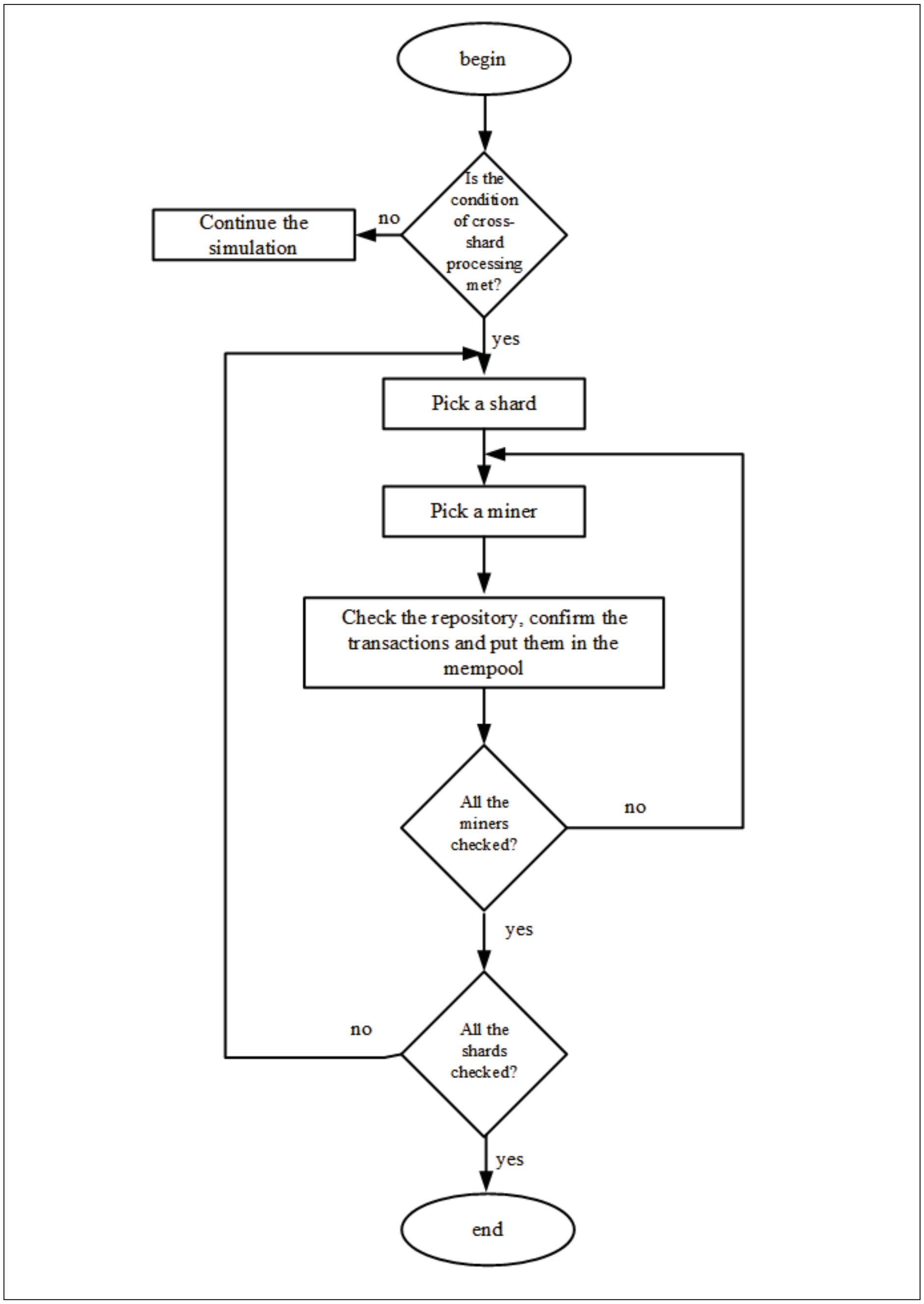

begin
Is the condition of cross-shard processing met?
no
Continue the simulation
yes
Pick a shard
Pick a miner
Check the repository, confirm the transactions and put them in the mempool
All the miners checked?
no
yes
All the shards checked?
no
yes
end

Figure 4.3 Flow chart of cross-shard transactions processing in the new simulator

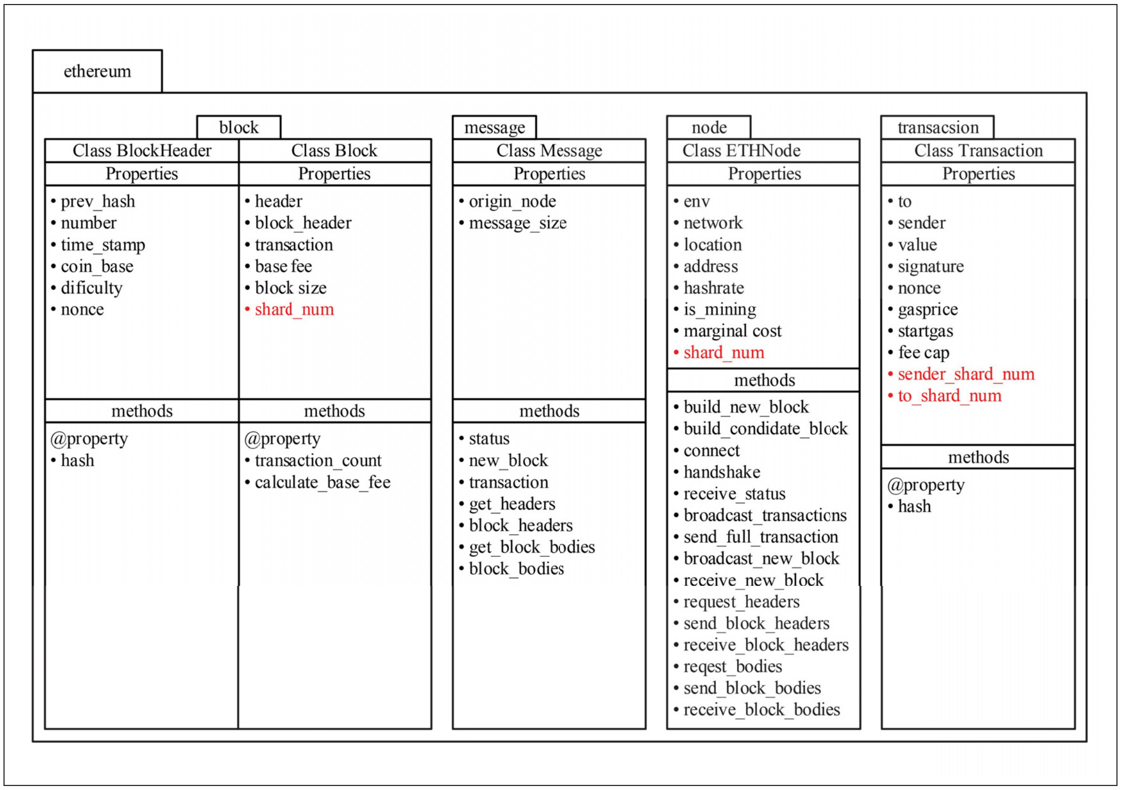


Figure 4.4 Class diagram of Ethereum node-level in the new simulator

4.3.1 60 nodes

During these experiments, we simulated the EIP-1559’s network with one, two, three, and four shards, respectively. It was only due to the lack of scalability that we would encounter during the experiment that we decided to perform this experiment with this number of shards. This parameter can be changed according to demand for other cases as well, depending on the number of shards. In the first experiment, in which there is only one shard, 448 blocks are generated. Fork probability is 0.99 and there are 219 orphan blocks in the network. As part of the last experiment, we had four shards. 21 blocks were created in the first shard, 19, 11 blocks in the second shard, and 16 blocks in the third shard. Therefore, 67 blocks are created in the sharded network as a whole. Moreover, the fork probabilities across the first to fourth shards are 0.26, 0.35, 0.16, and 0.405, implying that on average, the fork probability is 0.293.

As a result of this experiment, there are 9 orphan blocks for each shard, 4 orphan blocks for each shard, and 7 orphan blocks for each shard. Based on the comparison of the results of the sharded network with the non-sharded network, it is evident that the fork probability and the number of orphan blocks have decreased significantly. You can find the results obtained for the network with three and four shards in the tables 4.1 and 4.2.

Table 4.1 The results for a network with 60 nodes and 3 shards

Shard number	**Shard 1**	**Shard 2**	**Shard 3**
Fork probability	0.567	0.438	0.611
Number of orphan blocks	11	13	11
Network delay (ms)	700.889	684.176	674.448
Number of created blocks	24	33	27

Table 4.2 The results for a network with 60 nodes and 4 shards

Shard number	**Shard 1**	**Shard 2**	**Shard 3**	**Shard 4**
Fork probability	0.261	0.355	0.16	0.405
Number of orphan blocks	9	7	4	7
Network delay (ms)	375.94	411.224	147.53	387.8
Number of created blocks	21	19	11	16

As can be seen from the comparison of these two tables, the network with four shards has a lower fork probability than the network with three shards. The same applies to network delay, orphan blocks, and the total number of blocks created in each shard. We calculated the average results for all four experiments to obtain a more accurate estimate of the effect of sharding on fork occurrence probability. A summary of the results can be found in the table 4.3. It is evident from the table that increasing the number of shards results in a lower fork probability, which is also supported by the number of orphan blocks.

Figures 4.5, 4.6, and 4.7 provide graphs depicting fork probability, orphan block number, and network delay for 60 nodes, respectively. According to these Figures, increasing the number of shards has a positive impact on the likelihood of a fork.

Table 4.3 Average results for a network with 60 nodes

Number of shards	**1**	**2**	**3**	**4**
Average fork probability	0.99	0.88	0.538	0.293
Maximum fork probability	0.99	0.8995	0.611	0.405
Average number of orphan blocks	219	27	11.66	6.75
Maximum number of orphan blocks	219	29	13	9
Average network delay (ms)	11030.045	1433.115	686.501	330.62
Maximum network delay (ms)	11030.045	1462.48	700.88	411.224
Average number of created blocks	448	59	28	16.75
Maximum number of created blocks	448	63	33	21
Total number of created blocks	448	118	84	67

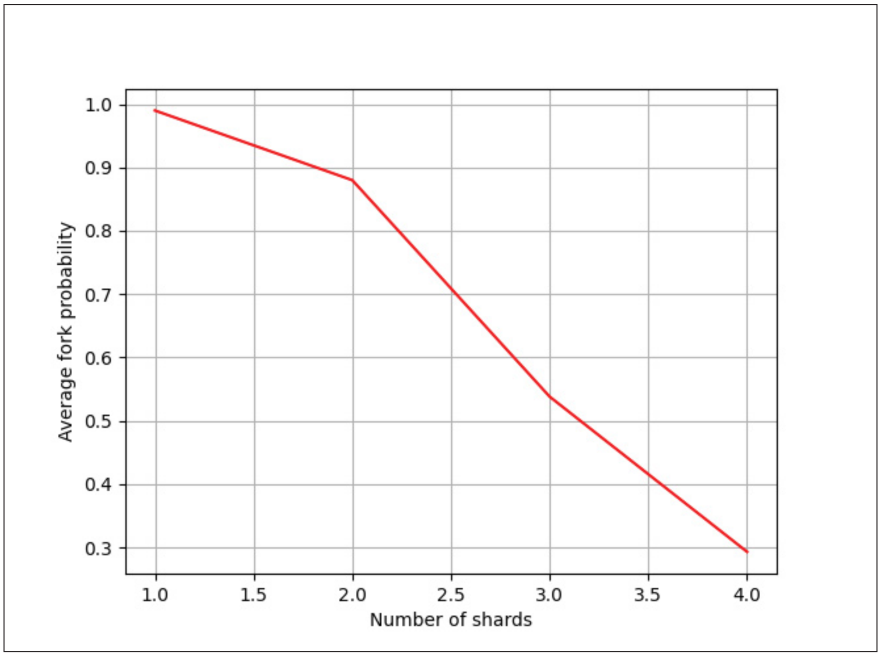


Figure 4.5 How fork probability changes in each by increasing the number of shards

4.3.2 120 nodes

In the tables 4.4 and 4.5, the results are presented for networks with three and four shards, respectively.

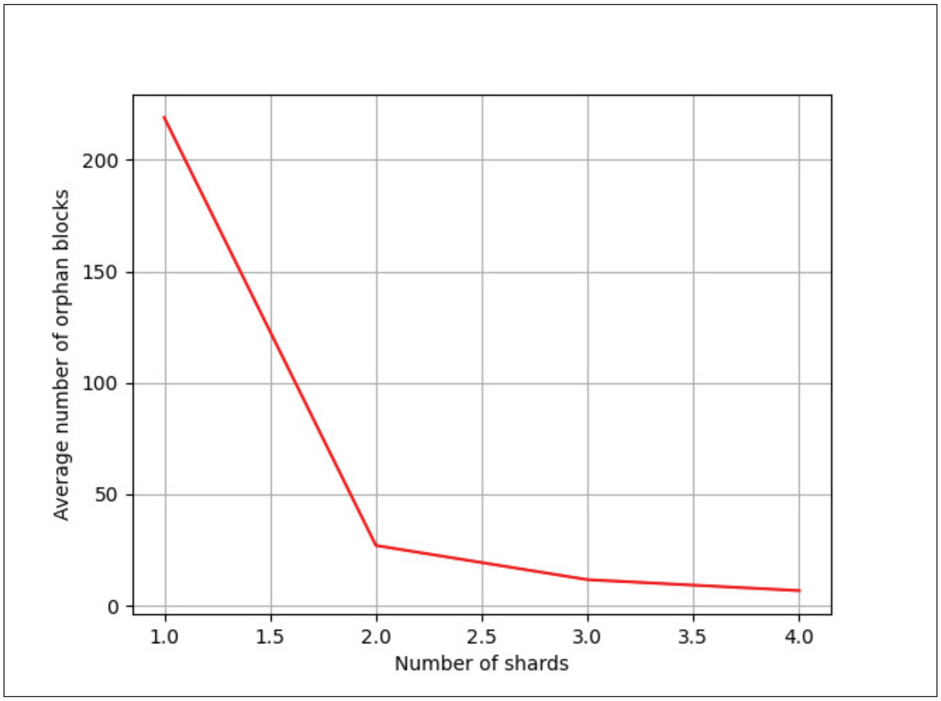


Figure 4.6 How number of orphan blocks changes by increasing the number of shards

There is no difference between the two tables in terms of the effect of adding a shard on the likelihood of forks occurring. On the basis of these tables, each shard in the second case has a lower level of network delay than the shards in the first case. In this way, the probability of a fork is reduced. Also shown in the table 4.6 are the average results of all experiments.

Table 4.4 The results for a network with 120 nodes and 3 shards

Shard number	**Shard 1**	**Shard 2**	**Shard 3**
Fork probability	0.88	0.69	0.86
Number of orphan blocks	10	12	17
Network delay (ms)	1443.228	1286.09	1409.702
Number of created blocks	23	28	37

According to the table, there are 67 blocks generated in the sharded network with four shards. Accordingly, each shard has an average of 16.75 blocks. Additionally, there are approximately 7.25 orphan blocks created per shard on average. As a result of these numbers, 16.75 blocks per shard and 7.25 orphan blocks per shard, the average fork probability is approximately 52%.

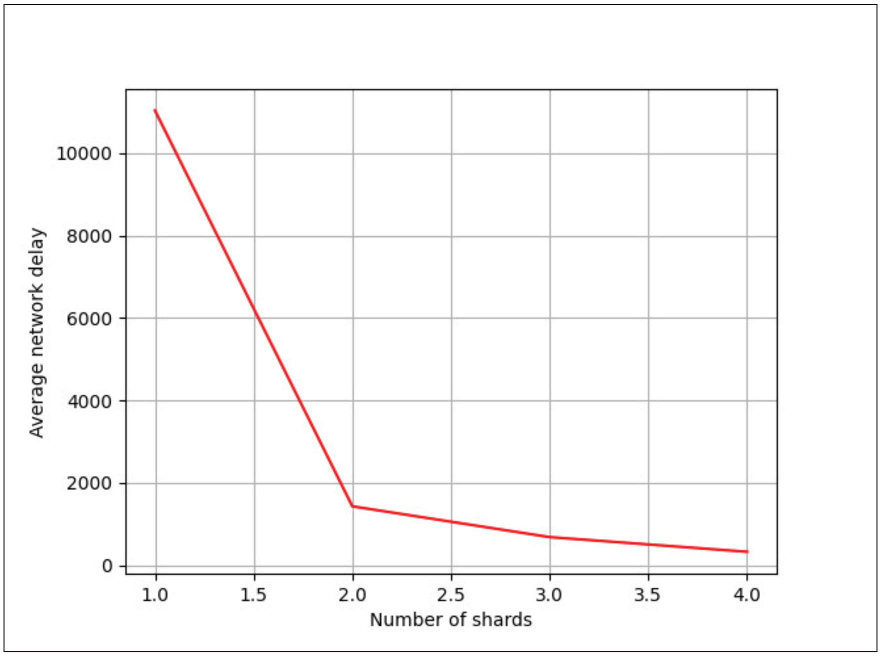


Figure 4.7 How network delay changes by increasing the number of shards

Table 4.5 The results for a network with 120 nodes and 4 shards

Shard number	**Shard 1**	**Shard 2**	**Shard 3**	**Shard 4**
Fork probability	0.508	0.433	0.66	0.521
Number of orphan blocks	4	7	7	11
Network delay (ms)	723.411	826.910	805.487	763.458
Number of created blocks	11	18	17	24

Alternatively, in a non-sharded network, the number of total blocks, the number of orphan blocks, and the fork probability are respectively 458, 217, and 99%. In Figures 4.8 through 4.10, graphs of the average fork probability, the average number of orphan blocks, and the average network delay are presented.

These Figures also indicate a decreasing trend in fork probability, orphan blocks, and network delays. This is due to the addition of more shards to the network. Further, as shown in table 4.3, increasing each shard results in a 23%, 61%, and 63% decrease in fork probability, orphan

Table 4.6 Average results for a network with 120 nodes

Number of shards	**1**	**2**	**3**	**4**
Average fork probability	0.99	0.9857	0.81	0.527
Maximum fork probability	0.99	0.99	0.88	0.66
Average number of orphan blocks	227	29	13	7.25
Maximum number of orphan blocks	227	39	17	11
Average network delay (ms)	22495.154	3097.77	1379.67	779.814
Maximum network delay (ms)	22495.154	3122.956	1443.221	826.91
Average number of created blocks	458	62	28	17.5
Maximum number of created blocks	458	84	37	24
Total number of created blocks	458	124	88	70

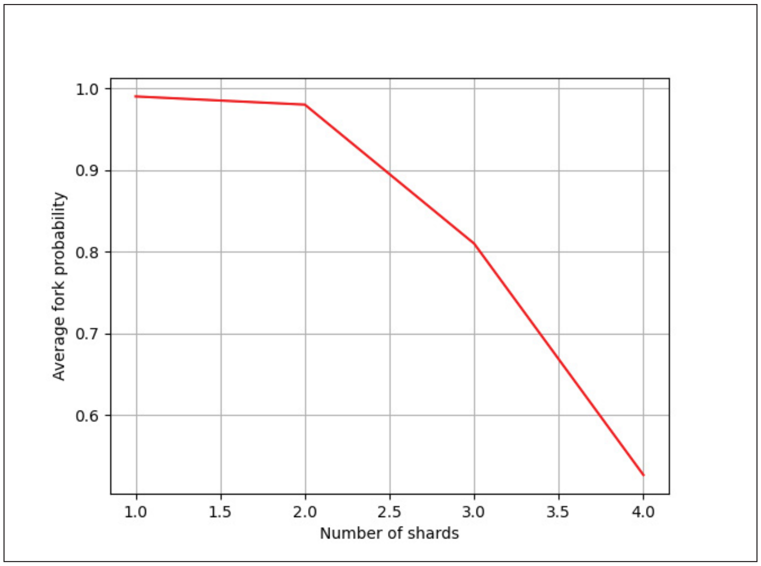


Figure 4.8 How fork probability changes in each by increasing the number of shards

blocks, and network delay, respectively. According to the table 4.6, these percentages are 15%, 62%, and 61%, respectively, for a network with 120 nodes.

According to the study, a reduction in the probability of a fork occurs as the number of shards increases. As a result, all of our simulations assume that the nodes are equally divided into different shards. As a result, all shards in our simulation contained the same number of nodes. There is no doubt that a shard with more nodes will have a higher chance of forking than one with

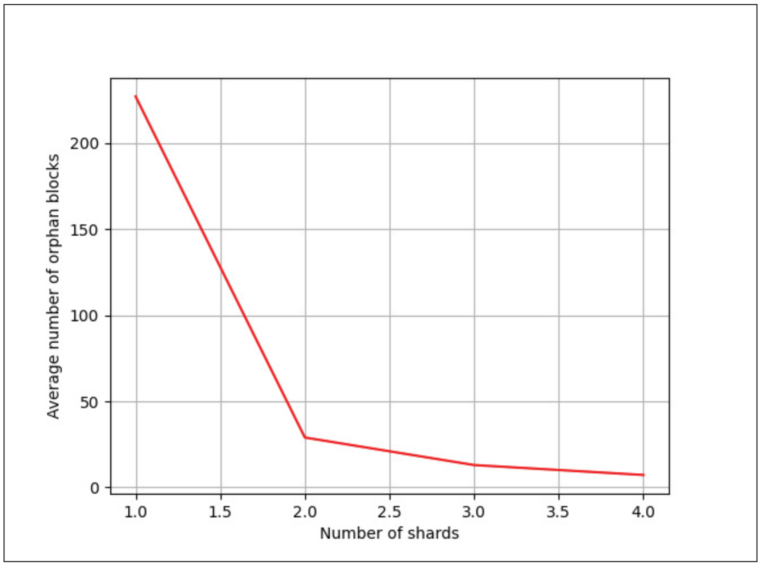


Figure 4.9 How number of orphan blocks changes by increasing the number of shards

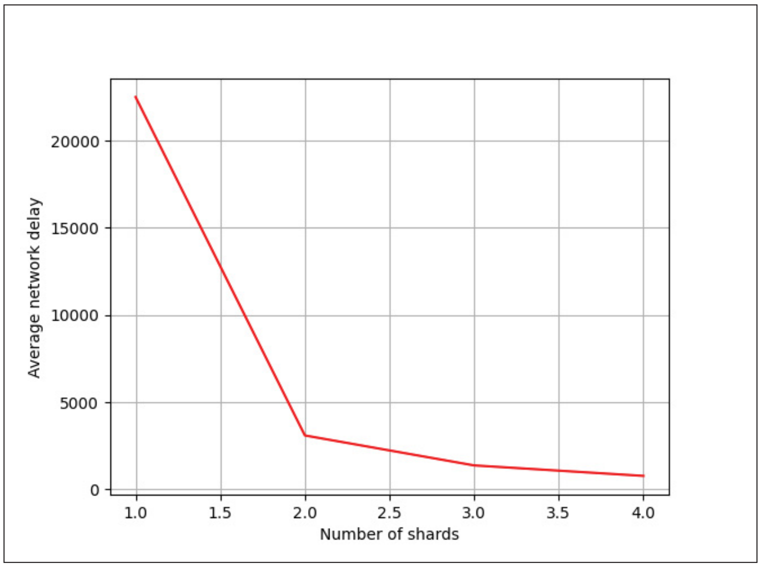


Figure 4.10 How network delay changes by increasing the number of shards

fewer nodes. Further, in order to achieve a reasonable conclusion about the effect of sharding on fork occurrence probability, *validation degree* and θ were set at 1 and 6, respectively.

4.4 Summary

This chapter presents a novel simulator that can simulate the EIP-1559 network as well as sharded networks. Using the proposed simulator, we investigated the impact of sharding on the probability of forking. This was achieved by simulating two networks for EIP-1559, each containing 60 and 120 nodes. Our experiments examined the effects of increasing the number of shards from one to four. A fork was more likely to occur, the number of orphan blocks increased, and the network was more likely to experience delays.

As a result of adding a shard to either network, the number of orphan blocks is reduced by 60% on average. There is a positive effect of sharding on the fork issue as demonstrated by the results. However, sharding increases the vulnerability of the network to 51% attacks. Therefore, in a sharded network, each shard has a reduced hash rate compared to a non-sharded network. In this way, attackers can more easily control shards rather than the entire network. It is therefore possible to examine the effect of increasing the number of shards on the vulnerability of the network against 51% attacks in future research.

CHAPTER 5

NEW ON-CHAIN GOVERNANCE MODEL BASED ON PARTICLE SWARM OPTIMIZATION FOR FORK REDUCTION

The content of this chapter concerns how sharding would affect the probability of forks. The new on-chain governance model has also been implemented that utilizes Particle Swarm Optimization in order to ensure that the number of forks between different shards are reduced. Section 5.1 presents various motivating use cases for the sharding solutions and on-chain governance methods to ensure the scalability. Section 5.2 explains the proposed new on-chain governance model for Ethereum network. The results obtained from our study give us the confidence that the proposed on-chain governance model reduces the risks associated with forking and maintains a positive user experience as a result of the results obtained. Section 5.3 presents those results. Finally, a conclusion is offered in Section 5.4.

5.1 Introduction

The use of blockchain technology has become increasingly popular since Bitcoin was introduced in 2008 (Nakamoto, 2008). The decentralized nature of this novel technology has made it a hot topic among researchers since then. Decentralization, transparency, immutability, and security are the most notable features of this revolutionary technology. There are a number of challenges associated with blockchain technology, as with other technologies. A major challenge in blockchain networks is the fork. When two (or more) miners propagate their blocks simultaneously, this occurs. The node in this situation will accept the block that arrives earlier as the tip of its chain. Thus, two (or more) branches of the chain will grow simultaneously. The conflict will be resolved by selecting the longest chain, and the remaining blocks will be considered orphans.

This can result in security issues for the network, a decrease in trust among users, and a decrease in the value of the coin as a result. A fork can be caused by a variety of factors, but network delays are the most common. Due to network delays and traffic, newly generated blocks may arrive at the nodes of the network sooner or later than each other. Accordingly, the last block of the nodes is determined by the arrival of the first block.

Another challenge associated with blockchain systems is their low transaction processing rate. A typical Ethereum network processes about 15 transactions per second, while other payment systems, such as VISA (Georgiadis, 2019; Gao *et al.*, 2019), are able to process thousands of transactions per second. With an increase in the number of nodes, this problem becomes more challenging to solve. A further challenge associated with blockchain networks is their scalability. The concept of sharding was introduced in Elastico (Luu *et al.*, 2016) in order to overcome this problem. Sharding involves dividing nodes into groups so that each group runs its own chain at the same time. Through the use of sharding, it is possible to process transactions more efficiently. In sharded networks, there are two types of transactions:

- Intra-shard transactions are transactions between nodes within a single shard
- Cross-shard transactions are transactions which are made between nodes across different shards

In a sharded network, transactions carried out between two nodes are confirmed in the same way as transactions carried out between nodes in a non-sharded network. In spite of this, for a cross-shard transaction in which the origin and destination are in different shards, both shards must approve the transaction in order for it to be included in the block. As a result, it takes a longer period of time to confirm these types of transactions.

In order to maintain a network's stability and security, a governance mechanism is used to control and govern it. We have introduced several governance models, which can be categorized into two main categories: on-chain governance and off-chain governance. An off-chain governance model involves developers making decisions outside of the blockchain, which is contrary to the very nature of decentralization associated with blockchain technology (Cao *et al.*, 2021). In the event of disagreements regarding off-chain governance, the network may undergo a hard fork.

The on-chain governance model allows users to make decisions through voting. As a result, these types of governance models are more secure and transparent. Based on particle swarm optimization algorithm, this research proposes a novel on-chain governance model to improve user experience and fork probability.

To illustrate this, we have added the ability to simulate a sharded network governed by the suggested on-chain governance model to our blockchain simulator. Our next step was to conduct several experiments to determine whether sharding affects the likelihood of a fork. Furthermore, we have applied our proposed on-chain governance models to two sharded networks with two and four shards, respectively, to test how well the proposed method reduces the fork probability and maintains a satisfactory state for users. In addition, we have chosen the EIP-1559 network as a case study for our analysis.

A summary of the main contributions of this research is as follows:

1. Investigating the effects of sharding on the likelihood of forks
2. An on-chain governance model based on particle swarm optimization is presented
3. Developing a cost function for optimizing the PSO algorithm based on fork probability and user experience

5.2 Proposed model

By utilizing the Particle Swarm Optimization algorithm (PSO) as an on-chain governance model, we are able to maintain a balance between fork probability and users' experience on the chain. This model controls the following parameters: *validation degree*, *miner's minimal marginal cost*, *average block size*, and *average time between blocks*.

In order to find the most suitable solution, multiple competing objectives must be met. One method for accomplishing this is the use of optimization algorithms that can address multiple objectives at the same time. In recent decades, metaheuristic algorithms have become increasingly popular for solving both single-objective optimization problems as well as multi-objective optimization problems Konak, Coit & Smith (2006); Rao, Rai, Ramkumar & Balic (2016). Meta-heuristic algorithms have a number of significant characteristics, including their independence from problems, ease of use, and ability to solve problems. Due to their independence from the problem, these algorithms do not take into account the specific characteristics of each problem.

A random node in the network is connected to each learner at the beginning of the learning process. This node obtains critical network parameters from the nodes that are connected to it. Afterwards, each learner runs a complete PSO independently to determine which parameters should be used to maintain a balance between fork occurrence probability and user experience. This is due to the fact that each learner experiences a proper balance between the probability of fork occurrence and the user experience.

As a result of the random connection between the learners and the other nodes, each of them has a different value for the parameter. This is due to the random connection between them. In a consensus mechanism, different solutions proposed by learners are aggregated. Based on the variety of solutions proposed by the learners, we are able to reach a final decision about the entire network. Detailed explanations of the newly developed simulator, the input parameters and their effects on the fork probability are provided in the following subsections. Furthermore, they provide a detailed explanation of the structure of particles and the cost function of the PSO algorithms.

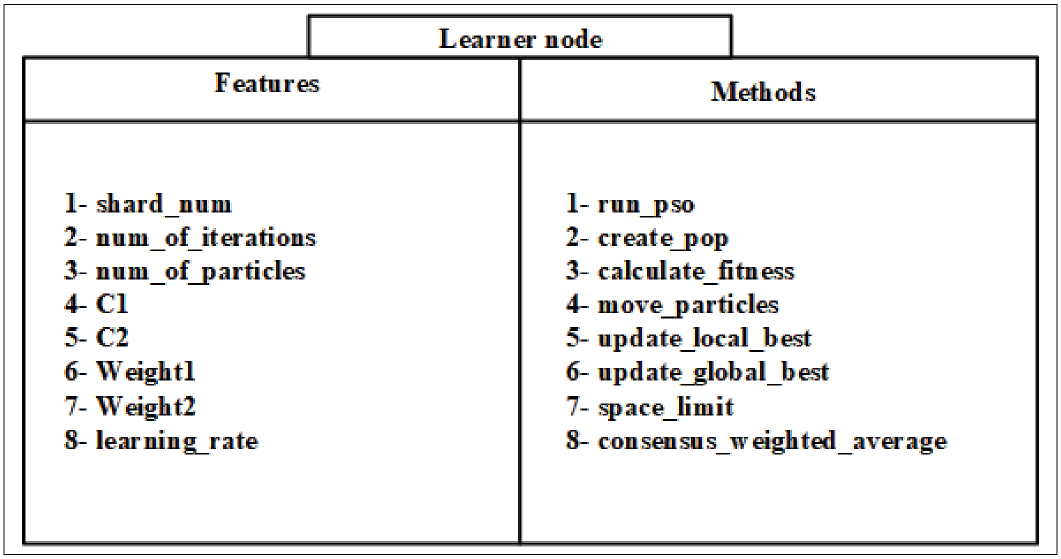
5.2.1 Improved simulator

A simulator would be required to evaluate the effectiveness of our proposed on-chain governance model. By adding the capability of simulating sharded networks to Blocksim, a simulation tool developed by the authors of (Faria & Correia, 2019), our work contributed to the improvement of the existing blockchain simulators. A PSO-based on-chain governance model was also developed and implemented, where our experiments were conducted.

In our simulator, we have added a newly-created node called *learners*. The responsibility of this node is to collect input parameters from their neighbors, run PSO, and provide an optimal solution based on the input parameters. At the beginning of every shard, a predetermined number of blocks are generated before these nodes become active. By analyzing the blocks that you have created, they provide you with the optimal solution. In order to reach a consensus, all suggestions are compiled using a consensus mechanism. A consensus mechanism is used to aggregate the solutions proposed by different learners in each shard so as to maintain control over the whole chain of solutions. A number of methods have been proposed in order to develop consensus mechanisms, including the (Zhang, Schmidt, White & Dubey, 2019a) method.

Throughout this study, we have used a weighted average mechanism to determine *validation degree*, *minimum marginal cost*, *average block size*, and *interval between blocks*. In this procedure, weights are assigned to each solution based on the sum of the hashes produced by all miners who are connected to the solution. The learner node with the highest weight is connected to the miners with the highest total hash rate. It is one of the learners associated with the miners with the highest total hash rate. Upon completion of the averaging process, the miners generate the *marginal cost* and *block size* randomly based on the average value determined by the governance model.

In addition, it is computed by averaging the values for *validation degree* and *average time between blocks* for each shard. It was necessary to take into account the parameter *learning_frequency* when developing a governance model. When determining when the governance model's tasks should be completed, this factor will be taken into account. Thus, if the governance model is *learning_frequency* = *4*, then it should be able to determine the optimal values for each of the four blocks. It is followed by the creation of four more blocks by the model. All simulations we have conducted have been run with *learning_frequency* set to 4. You can see this in Figure 5.1, which shows the class diagram of the learner nodes.



Learner node

Features	Methods
1- shard_num	1- run_pso
2- num_of_iterations	2- create_pop
3- num_of_particles	3- calculate_fitness
4- C1	4- move_particles
5- C2	5- update_local_best
6- Weight1	6- update_global_best
7- Weight2	7- space_limit
8- learning_rate	8- consensus_weighted_average

Figure 5.1 Class diagram of the learner nodes in the new simulator

5.2.2 Input parameters

This study focuses primarily on the occurrence of forks. As a result, it has been demonstrated that the proposed governance model reduces the likelihood of a fork occurring. Users may experience a negative user experience as a result of reducing the probability of forking.

A reduction in *miners' marginal cost* may decrease the probability of forking, however, this may result in overpayments for miners and an increase in the amount of time it takes for users to receive their blocks. As a result of the parameters selected, the following paragraphs provide an explanation of how the network behaves.

- The *Miners' minimum marginal cost* for EIP-1559 refers to the amount that a user must pay to the miner in order to have his transaction included in the block. Due to the higher marginal costs, users are required to pay more per transaction. Therefore, they are less likely to be included in the block. Consequently, the time it takes to create a block could increase. This will result in fewer blocks being created over time. A fork of the blockchain is less likely to occur as a result of this.

  Due to this effect, increasing the marginal cost of a product may adversely affect the user experience, which may negatively impact the customer's satisfaction. As a result of the above, this parameter negatively affects the user's experience. Fork probability and the user experience are directly influenced by this parameter, which is defined as our objective function. According to our simulations, we can generate a value between 0 and 0.9 for *miners' marginal cost*.

In order to determine the mean marginal cost, it is necessary to use the governance model. The average marginal cost can be calculated based on each run of the governance model that has been applied to the final solution. Once the average value of the marginal cost has been determined, the miners will draw random numbers from that value to calculate their marginal cost. Additionally, the premium value of each transaction is also generated at random between 0 and 1.

- *Validation degree (v)*: The number of nodes that need to validate a newly created block in order to be approved as the new tip of the chain is determined by this parameter. If v = 0.5, half of the nodes in the chain approve it, and once it has been verified, they add it to their chain. In contrast, the other half accepts it without verifying it. Therefore, once a block has been validated, when v = 1, all nodes should accept the block.

  This parameter is strongly correlated with the number of nodes involved in the process of block verification, resulting in an increase in network delay (see equation 2.1) as the primary cause of forks in the chain. As a result, the likelihood of a fork is higher when the *validation degree* of the text is high. Using the governance model, it was determined that the optimal value of *validation* is between 0.5 and 1.
- *Average block size (number of transactions in the block):* Based on the assumptions made in this research, the size of each block will be determined at random and around an average block size. Based on the length of the parameter, this parameter determines the size of the blocks. There is a direct correlation between the length of a block and the amount of time it takes for it to complete and propagate across the network. In light of this, it is likely that there will be a delay as well as a higher probability of a fork in the near future.

  In the simulations, a block can contain a maximum of 10 transactions and a minimum of 100 transactions. In accordance with the governance model, the governance model will be responsible for determining the optimal value of *average block size* between 10 and 100. It is the responsibility of the miners to determine the block size randomly somewhere around this average value.

- *Time between blocks* This parameter determines the minimum amount of time between consecutive blocks. It may be possible to reduce the probability of a fork by increasing the time interval between two consecutively created blocks. However, this can also result in an increase in network latency, which is not beneficial for users, as it increases the chances of forking. Therefore, in order for the system to function effectively, the optimal value for this parameter must be determined. The governance model searches for the optimal value between 5 and 30 seconds by looking at the solution space based on the results of the experiments.

During simulation, learner nodes can be used very effectively to control these parameters. To ensure a positive user experience, an acceptable probability of forking must be maintained.

5.2.3 Particles' structure and cost function

In this study, each particle is composed of a vector containing the parameters of the network. Among these parameters are the validating degree, the miners' marginal cost, the average block size, and the time between blocks. A critical aspect of the design process is the development of a cost function that can be used to model both increasing and decreasing fork patterns. Additionally, it is important to model the user's experience. As indicated in equation 5.1a, the cost function used in this study is shown below:

$$cost = \tag{5.1a}$$

$$w1 * (1 - e^{-(\frac{validation\_degree}{minmarginalcost+avg\_block\_size+average\_time\_blocks})})+ \tag{5.1b}$$

$$w2 * (min\_marginal\_cost + avg\_block\_size + avg\_time\_blocks) \tag{5.1c}$$

The first term of the function is used to model the increasing or decreasing movement pattern of fork probability (Kwon *et al.*, 2017; Aoki *et al.*, 2019). This second term is intended to control the user's experience, particularly overpayments.

Fork occurrence probability and user experience are mutually contradictory when properly modeled, and can be traded off. A reduction in network latency can be achieved by increasing the *average marginal cost*. Forks are therefore less likely to occur as a result.

Increasing it too much, however, will not be advantageous to users since they will have to pay a large amount per transaction. In equation 5.1c, the second term is intended to prevent overpayments. Moreover, *w1* and *w2* are two coefficients that allow us to focus on the most critical objective, depending on the network's conditions. In the case of reducing the fork probability, for example, we should choose a larger value for *w1* than for *w2*.

In order to emphasize the reduction of forks in our simulations, presented in the following section, we fixed *w1* and *w2* to 0.9 and 0.1, respectively. The flow chart of the proposed on-chain governance process is illustrated in the Figure 5.2.

5.3 Evaluation and results

One of the major objectives of this study is to investigate the effects of sharding and the proposed on-chain governance method on the fork probability issue. In order to achieve this, two scenarios were used to simulate the network of EIP-1559 with 120 nodes:

- In the absence of an on-chain governance mechanism to monitor sharding's impact on forking probability
- Based on the proposed on-chain governance model, an analysis is conducted of the impact of on-chain governance on fork occurrence probability

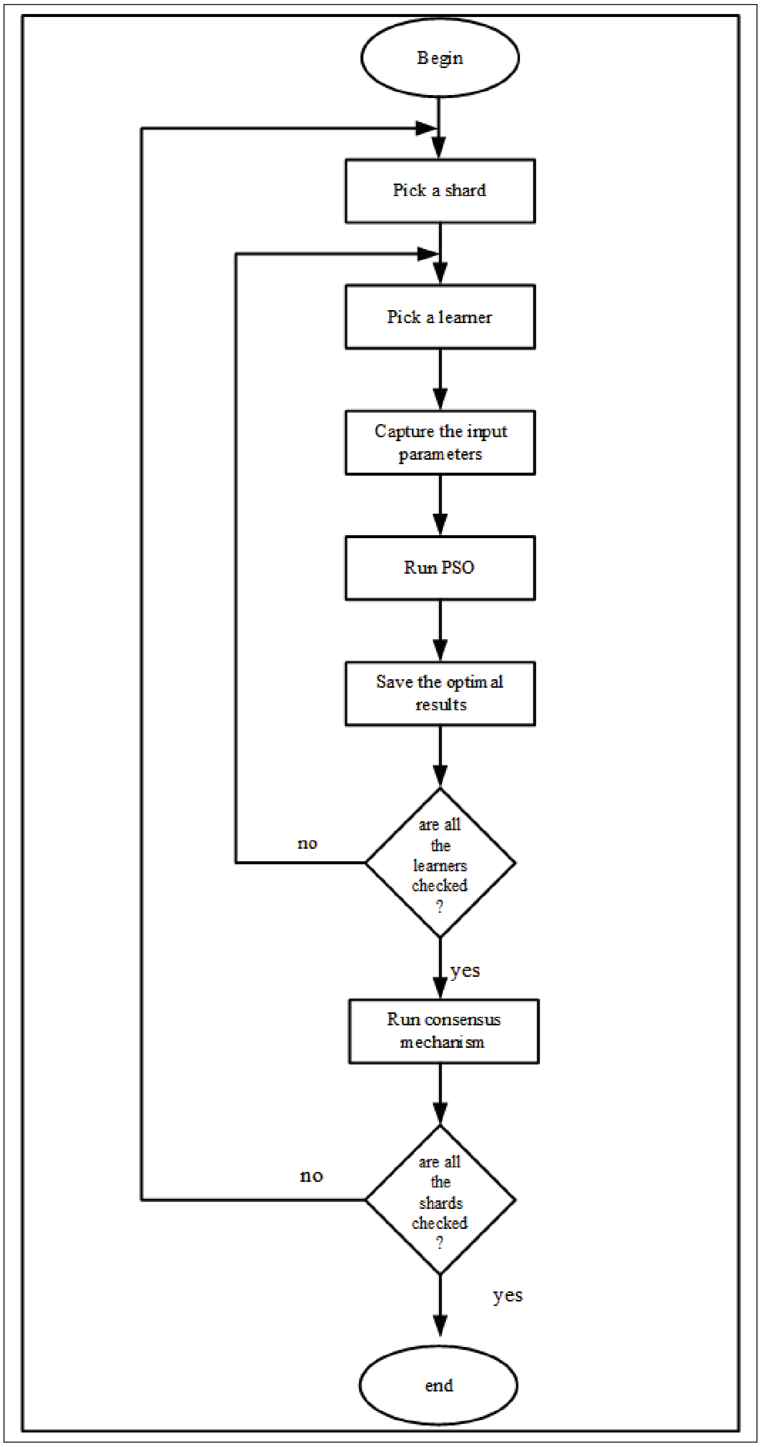


Figure 5.2 On-chain governance flow chart

5.3.1 Simulating a sharded network without on-chain governance

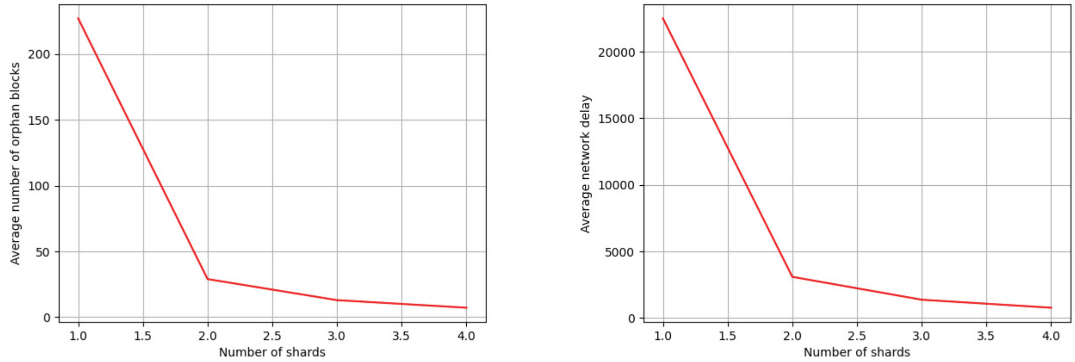
To investigate the effect of sharding on the fork issue, four simulations of EIP-1559's network were conducted using different numbers of shards, and key parameters were measured. In the table 5.1, it can be seen that the average results were achieved.

Table 5.1 Result for a network without on-chain governance with different shards

Number of shards	**1**	**2**	**3**	**4**
Average number of orphan blocks	227	29	13	7.25
Maximum number of orphan blocks	227	39	17	11
Total number of orphan blocks	227	58	39	29
Average network delay (ms)	22495.154	3097.77	1379.67	779.814
Maximum network delay (ms)	22495.154	3122.956	1443.221	826.91
Average number of created blocks	458	62	28	17.5
Maximum number of created blocks	458	84	37	24
Total number of created blocks	458	124	88	70

According to this table, a sharded network with four shards created 70 blocks in total. As a result, each shard consisted of an average of 17.50 blocks. Additionally, each shard contained an average of 7.25 orphan blocks. The total number of blocks created in a non-sharded network was 458, while the number of orphan blocks was 217. As shown in Figures 5.3a and 5.3b, the graphs show the average number of orphan blocks and the average network delay.

In addition, increasing the number of shards is likely to result in a reduction in the number of orphan blocks, as well as a reduction in the network delay. A comparison of the results without sharding with the results shown in Table 5.1 indicates that adding each shard results in a reduction of 62% in the number of orphan blocks and 61% in the number of network delays. It can be concluded from the results of the study that increasing the number of shards reduces the likelihood of forks occurring. Due to the increase in shards, this has occurred. In our simulations, we assume that every node is divided uniformly into different shards based on its configuration.



a) How number of orphan blocks changes by increasing the number of shards

b) How network delay changes by increasing the number of shards

Figure 5.3 Orphan blocks and network delay changes

As a result, in our simulations, all of the shards had the same number of nodes, regardless of their number. Forks are more likely to occur in shards with more nodes than in those with fewer nodes. A shard contains a significant number of nodes, which explains this phenomenon. Additionally, we decided to fix θ and *validation degree* to one in order to obtain a reasonable result regarding the effects of sharding on fork probability.

5.3.2 Simulating a sharded network with on-chain governance

The proposed on-chain model was evaluated by simulating four networks based on the EIP-1559 network: four networks with 1 to 4 shards, respectively. We have considered five learner nodes per shard in each case. A summary of the results is provided in the following subsections.

5.3.2.1 One shard

The purpose of this experiment was to examine only the effect of the proposed on-chain governance model on the occurrence of forks. The table 5.2 contains the results obtained in this experiment as well as those obtained when simulating the network without on-chain governance.

Table 5.2 The result of simulating a network with and without on-chain governance

Shards	Num orphan blocks	Total num blocks	Network delay
With on-chain governance	91	331	13169.88
Without on-chain governance	227	458	22495

The table 5.2 illustrates how well our on-chain governance model performs. According to this table, there is a difference in governance model between networks with and without on-chain governance. Thus, the number of orphan blocks has increased by the factor of 2.49, while the number of all created blocks has increased by the factor of 1.3. As a result, the number of orphan blocks has been dramatically reduced by implementing the proposed governance model. As shown in Figure 5.4, the proposed model functions as expected.

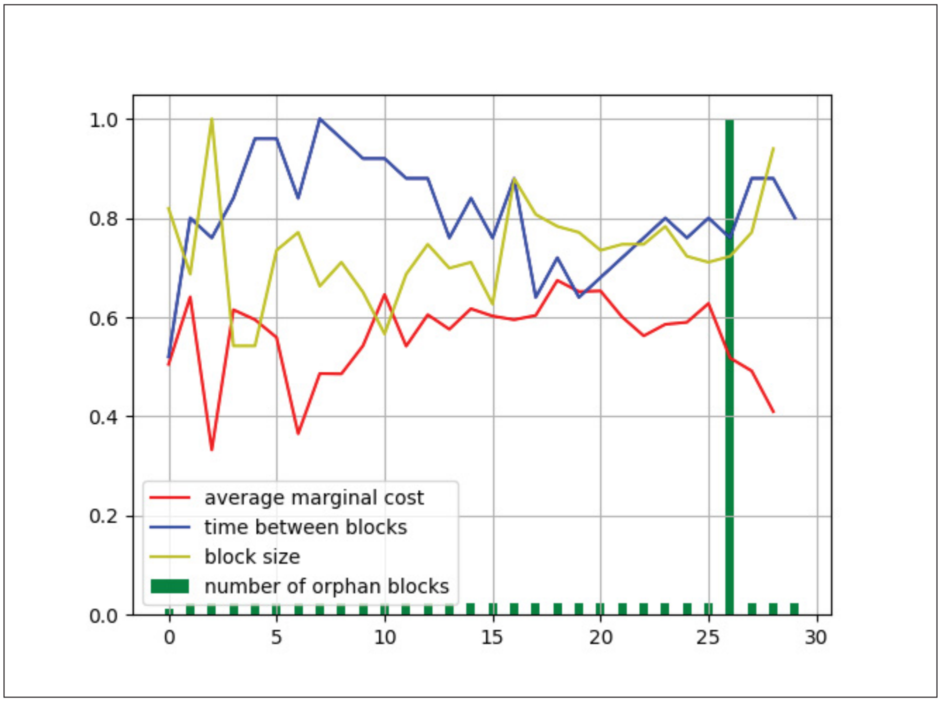


Figure 5.4 Performance of the governance model in a non-sharded network

An illustration of the normalized orphan block number between two consecutive runs of the proposed model is presented in the form of a bar graph. In addition, the other graphs illustrate the output of the model for *average marginal cost*, normalized *average time between blocks* and standardized *average block size*. Based on this Figure, it can be seen that the number of orphan blocks has increased significantly.

According to other graphs, this is due to a reduction in both *average marginal cost* and *average time between blocks*. There has been a reduction in the number of orphan blocks in the governance model as a result of this. It has resulted in an increase in *average time between blocks* and *average block size*, while *average marginal cost* has continued to decrease.

5.3.2.2 Two shards

The results of applying the proposed on-chain model are shown in Table 5.3.

Table 5.3 On-chain governance results on a sharded network with two shards

Shards	**Shard 1**	**Shard 2**
Validation degree	0.512	0.512
Minimum marginal cost	0.69	0.63
Block size	43.65	62.54
Time between blocks	21.79	19.16
Network delay (ms)	2198.85	1920.89
Number of created blocks	68	41
Number of orphan blocks	32	18

According to table 5.3, the on-chain governance model has fixed the validation degree at 0.512, which is basically the minimum value considered. Accordingly, the proposed method selects the most appropriate value for *validation degree* in order to reduce the likelihood of a fork. To ensure a balance between fork occurrence probability and user experience, the proposed model has chosen a moderate value for *minimum marginal cost*. In view of the fact that users will be expected to pay more for a straightforward transaction as the marginal cost increases, the likelihood of being included in a block will decrease. The creation of a block takes longer as a result.

In this way, the chances of forking are reduced to a considerable extent. In spite of the fact that *marginal cost* may be beneficial to users, it may also be detrimental to them. According to a study conducted by the University of British Columbia, a governance model has been developed in order to come up with a solution that will benefit both users and miners. According to the proposed model, the time between blocks for the first and second shards should be 21.79 seconds and 19.16 seconds, respectively, according to the proposed model. As can be seen once again, the model has chosen a moderate value for this parameter. Furthermore, decreasing *time between blocks* is also more beneficial to users in addition to increasing the probability of forks occurrence. In addition, the on-chain model found an average block size of 46.65 for the first shard and 62.54 for the second shard for *block size*.

It is clear from the values selected for this parameter that the model attempts to take into account alternative costs that are in conflict with each other. For the first shard, this resulted in 32 orphan blocks and for the second shard, this resulted in 18 orphan blocks. Thus, in the simulations we have performed as a result of this, 50 orphan blocks have been generated. As mentioned in the previous experiment, there were 58 orphan blocks in the previous experiment without governance models. There is also a slight advantage of sharded networks in terms of network delay, as they have a lower average network delay than non-sharded networks, which have an average network delay of 3097.77 milliseconds (ms). In Figures 5.5 to 5.6b, the governance model performed well in determining the *average marginal cost*, *average time between blocks*, and *average block size*.

In Figure 5.5, the red and green graphs illustrate how the governance model balances the fork probability with the user experience. It is evident from the figure that at certain points, the graphs exhibit an ascending pattern, while at other points, they exhibit a decreasing pattern, as it can be seen from the figure. Consequently, the proposed model sometimes increases or decreases *marginal cost* in order to reduce fork probability, and sometimes increases or decreases it in order to improve the user experience. Observations made by connected nodes can be used to determine whether the number of learner nodes should be increased or decreased in order to achieve the best results.

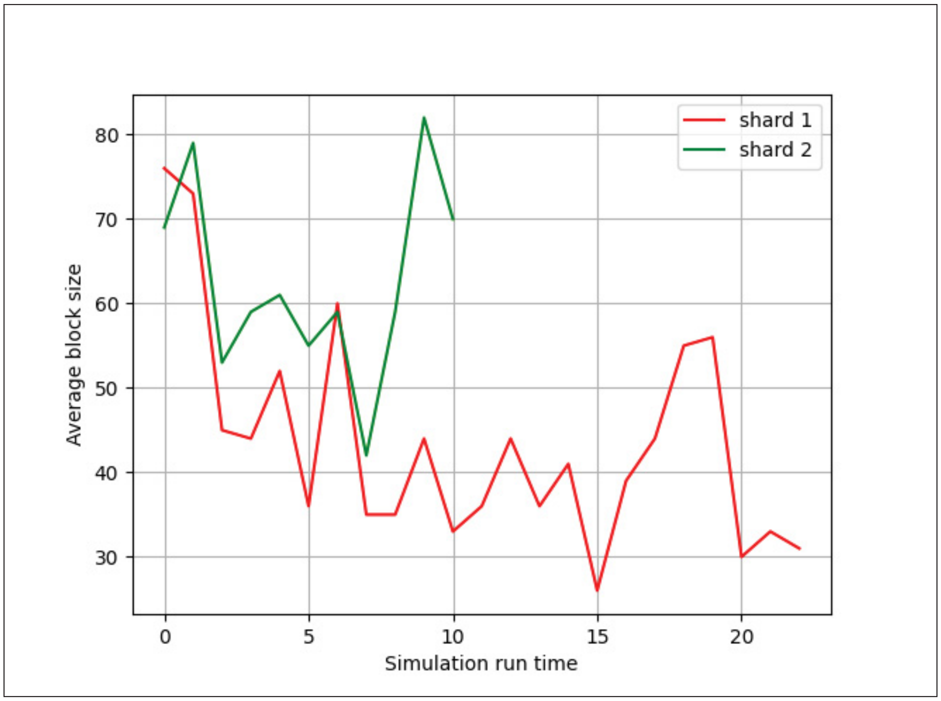
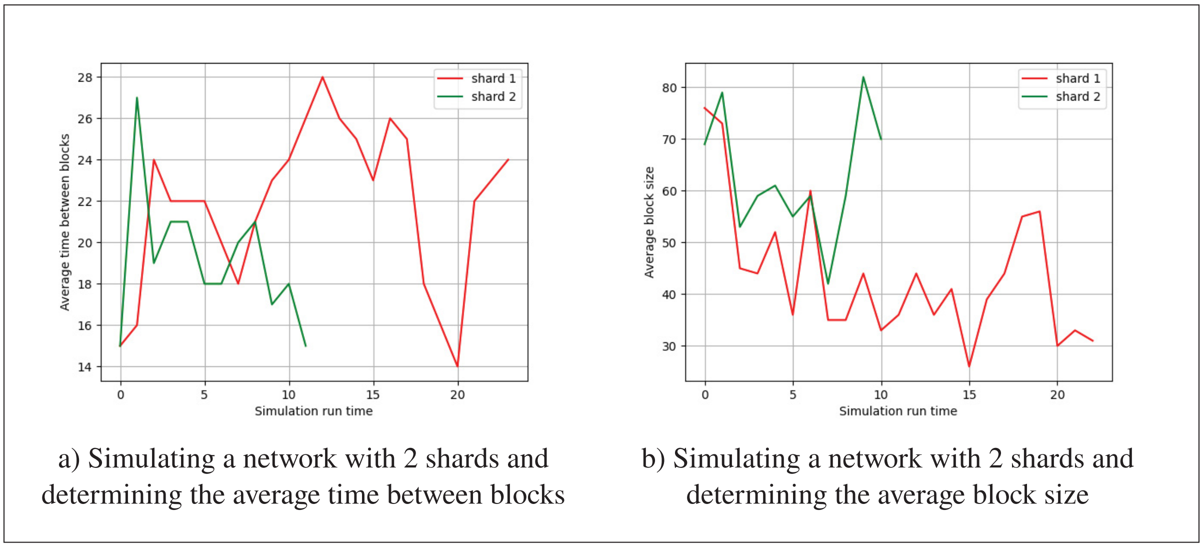


Figure 5.5 Simulation of a two-shard network: how the governance model calculates average marginal costs



a) Simulating a network with 2 shards and determining the average time between blocks

b) Simulating a network with 2 shards and determining the average block size

Figure 5.6 Orphan blocks and network delay changes

Both 5.6a and 5.6b use the same concept as *time between blocks* and *block size*. The governance model may decide to increase or decrease these parameters based on the results of monitoring the network from its connected nodes. The purpose of this is to reduce the likelihood of forks or to improve the user experience. The green graphs appear to be shorter than the red graphs based on these Figures. Due to the lower number of blocks created in the second shard as compared to the first shard, this is the case.

5.3.2.3 Three shards

Table 5.4 contains the results achieved for this experiment.

Table 5.4 On-chain governance results on a sharded network with three shards

Shards	**Shard 1**	**Shard 2**	**Shard 3**
Validation degree	0.57	0.5	0.5
Minimum marginal cost	0.65	0.415	0.383
Block size	67.5	76.6	81.6
Time between blocks	17	19.66	20.5
Network delay (ms)	444.736	470.435	478.751
Number of created blocks	19	22	27
Number of orphan blocks	15	10	12

According to this table, both shards have a minimum number of orphan blocks and all of their parameters are moderate. There is a direct relationship between average marginal cost, average time between blocks, and average block size, as shown in 5.7 to 5.9.

As can be seen in Figure 5.7, the marginal cost graphs for all shards show a decreasing trend. Even though this is beneficial for users, it may result in a much higher probability of a fork. As a result, the governance model has increased the *time between blocks* in order to maintain this trade-off and to prevent chains from forking, as illustrated in Figure 5.8.

5.3.2.4 Four shards

In Table 5.5, the results obtained following the implementation of the on-chain governance model on a network with four shards are presented. According to this table, the first shard creates the least number of orphan blocks. Among this shard, the *average marginal cost]* is the highest.

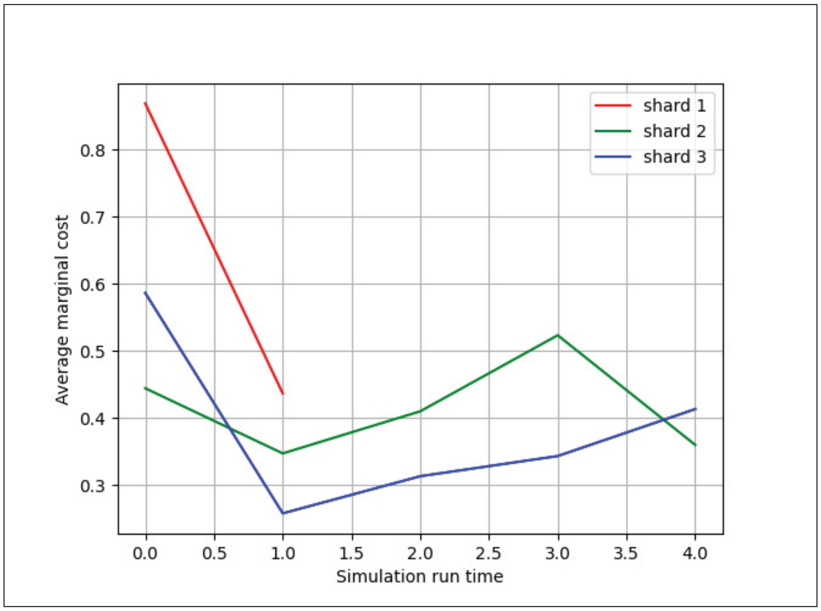


Figure 5.7 Simulation of a three-shard network: how the governance model calculates average marginal costs

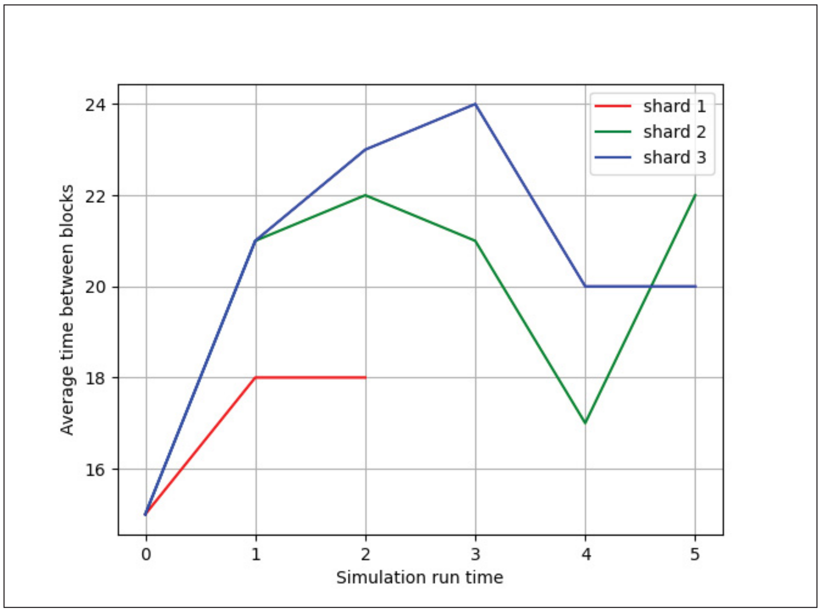


Figure 5.8 Simulation of a three-shard network: how the governance model calculates average time between blocks

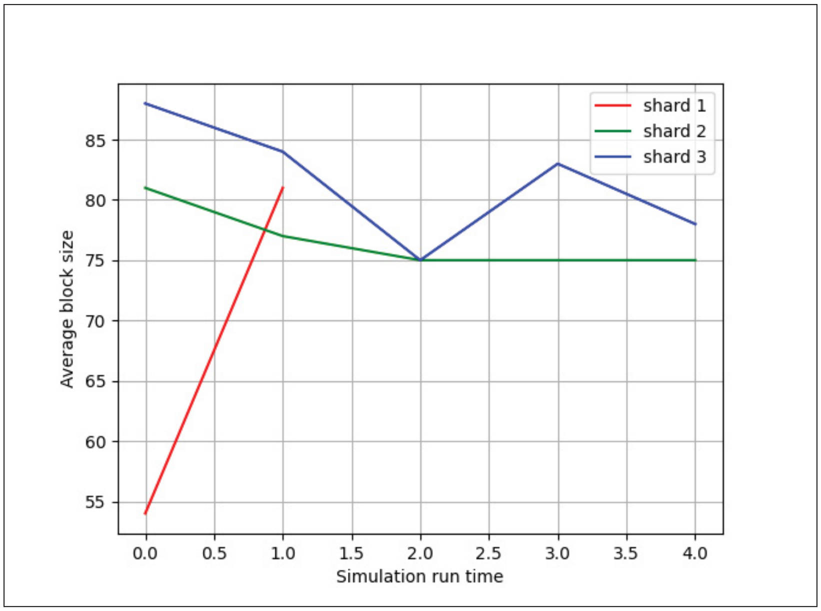


Figure 5.9 Simulation of a three-shard network: how the governance model calculates average block size

Table 5.5 On-chain governance results on a sharded network with four shards

Shards	**Shard 1**	**Shard 2**	**Shard 3**	**Shard 4**
Validation degree	0.519	0.5	0.5	0.5
Minimum marginal cost	0.623	0.535	0.485	0.579
Block size	59.5	50	62.125	61.63
Time between blocks	19.4	23.72	21.55	19.75
Network delay (ms)	521.72	634.91	576.44	596.27
Number of created blocks	11	20	16	22
Number of orphan blocks	4	9	7	10

The result will be that users will be required to pay a higher fee to have their transactions included in a block in this shard. Consequently, there is a reduced likelihood of being included in a block in this situation. In this shard, fewer blocks are created, and there is less network delay than in other shards. Furthermore, this shard has a higher *validation degree* than other shards, resulting in a lower fork probability and a higher *marginal cost*. Thus, this shard has the lowest forking probability of all the shards.

In the second and fourth shards, the number of orphan blocks is very similar to that of blocks created in the first shard. The average marginal costs of these two shards are almost equally different. When comparing *average block size* with *average time between blocks*, there is a significant difference. The second shard has a smaller *average block size* than the fourth shard, however the fourth shard has a longer *average time between blocks*. These parameters have a similar impact on the probability of fork occurrences and the user experience. It will result in longer block creation times if both of these factors are increased. Forks will be less likely to occur as a result of this.

Compared with the other shards, the third has a lower *average marginal cost*, but the *average block size* is the largest. As a result, although the cost of users in this shard is lower than in the other shards, they have to wait a longer period of time before their transactions are included in a block. Compared to other shards, this shard takes a longer time to create blocks. Furthermore, there was a significantly more significant difference between this shard and the first and fourth shards regarding *average time between blocks*.

As compared to the first shard, this shard has a lower*average marginal cost*, which is more beneficial to the users. However, it increases the likelihood of a fork in the future. Meanwhile, it has a larger *average block size*, which is inversely proportional to the likelihood of a fork. Although the third shard does not have the minimum fork probability, it is more cost-effective and provides users with a better waiting time and cost situation. The governance model's performance in determining *average marginal cost for four shards* and *block size for four shards* for each shard during simulations has been demonstrated in Figures 5.10 through 5.12.

According to Figure 5.10, the *average marginal cost* of the first shard decreases initially and then increases. It has been observed that lowering the *marginal cost* increases the fork probability, and vice versa. As shown in 5.12, the *average block size* graph of the first shard completely opposes its *marginal cost* graph. It begins with an increase and then decreases.

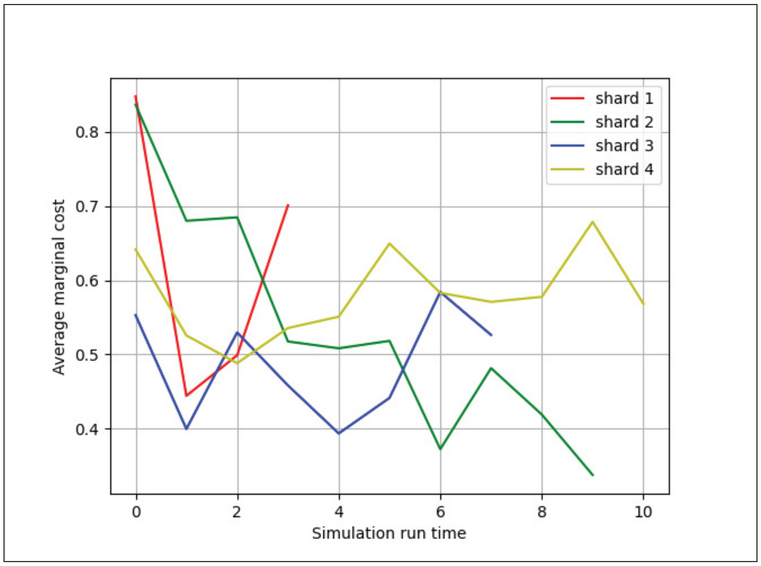


Figure 5.10 Simulation of a four-shard network: how the governance model calculates average marginal costs

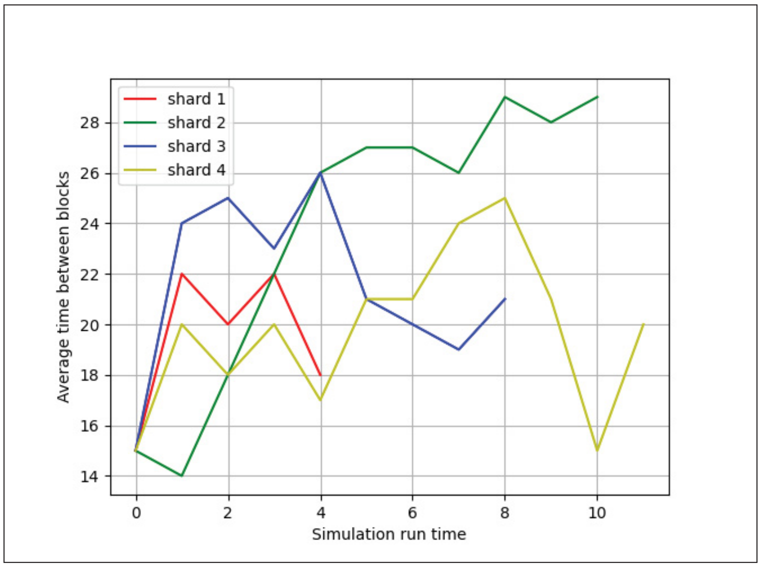


Figure 5.11 Simulation of a four-shard network: how the governance model calculates average time between blocks

Consequently, the on-chain governance model increased *average block size* to manage the increase in fork probability while maintaining a moderate range for *average block interval*.

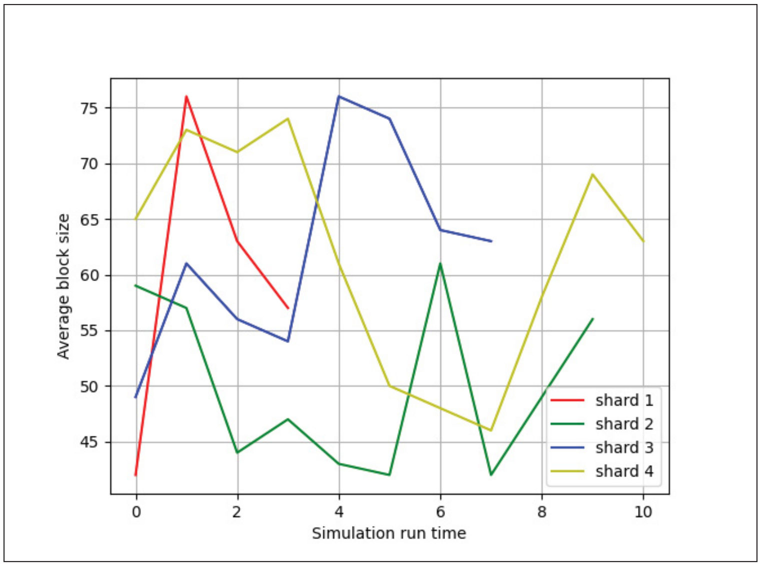


Figure 5.12 Simulation of a four-shard network: how the governance model calculates average block size

Even though *average marginal cost* fluctuates, it generally shows a downward trend, increasing the likelihood of a fork. According to 5.11, its *average time between blocks* displays an increasing pattern, which is inversely related to fork probability.

Alternately, reducing *average marginal cost* would improve the users' experience, but would increase the likelihood of a fork. Due to this issue, the on-chain governance model has increased the *average time between blocks*. For the *average block size*, according to Figure 5.12, the graph for the second shard does not exhibit any specific characteristics. As a result, it falls within a moderate range of values.

There appears to be a conflict between *average marginal cost* and *average block size* for the third shard. As shown in the Figures 5.10 and 5.12, when *average marginal cost* increases, on the other hand, *average block size* decreases, and vice versa. Additionally, the graph for *average time between blocks* for the third shard does not appear to have a clear pattern.

The Figures 5.10 and 5.12 illustrate the same concept for the fourth shard. A moderate range of values has been attempted for the parameters of the proposed method. The purpose of this is to maintain a balance between fork probability and the experience of the users of this shard.

5.4 Summary

It is the main objective of this research to compare the probability of forking between the proposed on-chain governance model and the impact of sharding, as well as the impact of sharding on the forking rate. In order to achieve this goal, we have simulated a network with 120 nodes and different numbers of shards according to the EIP-1559 specification. We increased the number of shards from one to four during our experiment. In order to determine the effects of this on the number of orphan blocks and the overall network delay, this study is being conducted. The addition of a shard to a network reduces the number of orphan blocks on average by 60% on average for both networks.

The results of this chapter indicate that sharding has a positive impact on the forking process. Sharding can, however, increase the network's vulnerability by as much as 51%. Sharded networks are characterized by having a lower hash power within each shard than non-sharded networks. Consequently, attackers are more likely to gain control over individual shards of a network than to gain control over the entire network.

Secondly, the proposed on-chain governance model was applied to the sharded network in order to evaluate its efficiency and impact on the likelihood of forks occurring. In order to achieve this objective, we simulated two networks. In order to accomplish this goal, these networks consisted of two and four shards, respectively. This chapter demonstrated that the proposed on-chain model was capable of solving the corresponding optimization problem and determining the optimal parameter settings for the network as a means of maintaining a balance between the fork probability and the user experience, based on the results obtained.

CONCLUSION AND RECOMMENDATIONS

As a conclusion, Section 6.1 provides a summary of this thesis, followed by Section 6.2 which provides an outlook on current and future research.

6.1 Summary

Public blockchain forks, one of the most significant issues, have not received sufficient attention. In order to provide a scalable platform for using blockchain technology in everyday life, much effort is being made to solve the scalability problem of existing blockchains. To reduce the effects of forks, this thesis concludes that a flexible and adaptable blockchain structure can provide the most optimal structure under a variety of circumstances. This can reduce fork probability to maximize scalability. A number of decentralized methods for governance and governance of blockchains have been developed in recent years as blockchain technology has progressed and developed. As a result, it may be challenging to reach consensus when determining and updating the optimal structure along this path. Decentralized methods can be used to address these problems. Three concrete contributions are presented in this thesis, focusing on the development of a dynamic blockchain architecture that utilizes a multi-objective PSO to solve a fork probability optimization problem in a dynamic environment.

Chapter 3 discusses fork modeling and its key parameters. In order to improve security, speed, and efficiency, a blockchain network should minimize the likelihood of forking. Parameter optimization for this purpose requires a thorough understanding of the behavior of blockchain systems. This study proposes an innovative fork model that incorporates previously unconsidered parameters, such as network delay and degree of validation. In order to verify the validity of our proposed method, several experiments were performed using the blockchain simulator on the Ethereum network and EIP-1559. This study determined that by lowering the validation degree and increasing the miner's marginal cost (as in EIP-1559), approximately 10% of the probability of a fork can be reduced. In addition, our method is highly accurate in predicting the probability of a fork based on the results of the experiments.

Chapter 4 examines the impact of sharding on fork probability. Sharding was designed to deal with the problem of low processing rates, but it has also been demonstrated to increase the scalability of the network as well. In order to determine whether adding new shards to a blockchain will affect the likelihood of forks, our study evaluates the impact of adding new shards to a blockchain. For the purpose of simulating sharded networks, a novel simulator has been developed. As a second step, we examined whether sharding affects the frequency of forks. Two EIP-1559 enabled networks with 60 and 120 nodes were the subject of several experiments. As a result of our study, we found that adding one shard reduces orphan blocks by 60%. For networks with 60 and 120 nodes, we propose a fork probability model that results in reductions of 23% and 15%, respectively.

Last but not least, the purpose of Chapter 5 is to extend Chapter 4, which aims to determine how sharding will affect forks as a result of sharding since sharding has the ability to solve this problem. Several experiments have been conducted using 120 nodes in the network EIP-1559 in order to achieve this objective. On average, adding a shard to the system reduced the number of orphan blocks by 60%. A novel method of particle swarm optimization has also been implemented to reduce the likelihood of forks between different shards. Based on the results of our study, we are confident that the proposed on-chain governance model mitigates the risks associated with forking and maintains a positive user experience.

6.2 Future work

A number of future research projects are highlighted in different directions. Future studies would focus on other indicators, such as performance, decentralization, or even security. Moreover, the presented model can be used to develop a multi-objective optimization function to provide a balance point for the blockchain trilemma (scalability, decentralization, and security). Further investigation of the proposed model for other consensus methods, such as the proof of stake (POS) method, would be beneficial in the future. It is also possible to improve the performance of the system through the use of other artificial intelligence methods, such as neural networks or machine learning models.

AUTHOR'S PUBLICATIONS

During the course of his PhD research, the author contributed to the following published and submitted research articles.

[1] Reza Nourmohammadi, Kaiwen Zhang. "Modeling the Fork Probability of Blockchains: Did EIP-1559 Improve Ethereum?", The Fourth International Conference on Blockchain Computing and Applications (BCCA), (2022).

[2] Reza Nourmohammadi, Kaiwen Zhang. "Sharding and Its Impact on Fork Probability." IEEE Global Emerging Technologies Conference (IEEE GET) (2022).

[3] Reza Nourmohammadi, Kaiwen Zhang. "New On-Chain Governance Model based on Particle Swarm Optimization for Fork Reduction." IEEE Access Journal (2022).

BIBLIOGRAPHY

Aazam, M. & Huh, E.-N. (2016). Fog computing: The cloud-iot\/ioe middleware paradigm. *IEEE Potentials*, 35(3), 40–44.

Abbate, T., Cesaroni, F., Cinici, M. C. & Villari, M. (2019). Business models for developing smart cities. A fuzzy set qualitative comparative analysis of an IoT platform. *Technological Forecasting and Social Change*, 142, 183–193.

Abosaif, A. N. & Hamza, H. S. (2020). Quality of service-aware service selection algorithms for the internet of things environment: A review paper. *Array*, 8, 100041.

Aggarwal, S. & Kumar, N. (2021). Hyperledger. In *Advances in Computers* (vol. 121, pp. 323–343). Elsevier.

Aiyar, K., Halgamuge, M. N. & Mohammad, A. (2021). Probability distribution model to analyze the trade-off between scalability and security of sharding-based blockchain networks. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6.

Al-Bassam, M., Sonnino, A., Bano, S., Hrycyszyn, D. & Danezis, G. (2017). Chainspace: A sharded smart contracts platform. *arXiv preprint arXiv:1708.03778*.

Al-Hawawreh, M., Moustafa, N., Garg, S. & Hossain, M. S. (2020). Deep Learning-enabled Threat Intelligence Scheme in the Internet of Things Networks. *IEEE Transactions on Network Science and Engineering*, 8(4), 2968–2981.

Al-Rubaye, S., Al-Dulaimi, A. & Ni, Q. (2019a). Power interchange analysis for reliable vehicle-to-grid connectivity. *IEEE Communications Magazine*, 57(8), 105–111.

Al-Rubaye, S., Rodriguez, J., Al-Dulaimi, A., Mumtaz, S. & Rodrigues, J. J. (2019b). Enabling digital grid for industrial revolution: self-healing cyber resilient platform. *IEEE Network*, 33(5), 219–225.

Alharbi, H. & Hussain, A. (2015). An Agent-Based Approach for Modelling Peer to Peer Networks. *2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim)*, pp. 532–537.

Alharby, M. & Van Moorsel, A. (2019). Blocksim: a simulation framework for blockchain systems. *ACM SIGMETRICS Performance Evaluation Review*, 46(3), 135–138.

AlZain, M. A. & et al. (2011). Mcdb: using multi-clouds to ensure security in cloud computing. *In DASC'11*, pp. 784–791.

Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., McCallum, P. & Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100, 143–174.

Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.".

Antoun, J., Kabir, M. E., Moussa, B., Atallah, R. & Assi, C. (2020). A Detailed Security Assessment of the EV Charging Ecosystem. *IEEE Network*, 34(3), 200–207.

Aoki, Y. & Shudo, K. (2019). Proximity neighbor selection in blockchain networks. *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 52–58.

Aoki, Y., Otsuki, K., Kaneko, T., Banno, R. & Shudo, K. (2019). Simblock: A blockchain network simulator. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 325–329.

Arribas, I., Arroyo, D. & Reshef Kera, D. (2020). Sandbox for minimal viable governance of blockchain services and daos: Claudia. *International Congress on Blockchain and Applications*, pp. 24–30.

Atik, J. & Gerro, G. (2018). Hard forks on the Bitcoin blockchain: reversible exit, continuing voice. *Stan. J. Blockchain L. & Pol'y*, 1, 24.

Atzei, N., Bartoletti, M. & Cimoli, T. (2017). A survey of attacks on ethereum smart contracts (sok). *International conference on principles of security and trust*, pp. 164–186.

Au, M. H., Liu, J. K., Fang, J., Jiang, Z. L., Susilo, W. & Zhou, J. (2013). A new payment system for enhancing location privacy of electric vehicles. *IEEE transactions on vehicular technology*, 63(1), 3–18.

B, I. & et al. (2008). Approximation algorithms for data placement problems. *SIAM Journal on Computing*, 38(4), 1411–1429.

Bae, J. & Lim, H. (2018). Random mining group selection to prevent 51% attacks on bitcoin. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 81–82.

Bai, J. & Hao, R. (2019). Comment on "Privacy-preserving public auditing for non-manager group shared data". *The Journal of Supercomputing*, 100(4), 1277–1294.

Bao, K., Valev, H., Wagner, M. & Schmeck, H. (2018). A threat analysis of the vehicle-to-grid charging protocol ISO 15118. *Computer Science-Research and Development*, 33(1-2), 3–12.

Bao, Z., Wang, K. & Zhang, W. (2019). An auditable and secure model for permissioned blockchain. *Proceedings of the 2019 International Electronics Communication Conference*, pp. 139–145.

Bayram, I. S. & Papapanagiotou, I. (2014). A survey on communication technologies and requirements for internet of electric vehicles. *EURASIP Journal on Wireless Communications and Networking*, 2014(1), 1–18.

Baza, M. & et al. (2021). Privacy-preserving blockchain-based energy trading schemes for electric vehicles. *IEEE Transactions on Vehicular Technology*, 70(9), 9369–9384.

Beck, R., Müller-Bloch, C. & King, J. L. (2018). Governance in the blockchain economy: A framework and research agenda. *Journal of the Association for Information Systems*, 19(10), 1.

Benisi, N. Z., Aminian, M. & Javadi, B. (2020). Blockchain-based decentralized storage networks: A survey. *Journal of Network and Computer Applications*, 102656.

Bentov, I., Lee, C., Mizrahi, A. & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review*, 42(3), 34–37.

Blum, J. J. & Eskandarian, A. (2007). A reliable link-layer protocol for robust and scalable intervehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 8(1), 4–13.

Bodet, C., Schülke, A., Erickson, K. & Jabłonowski, R. (2012). Optimization of charging infrastructure usage under varying traffic and capacity conditions. *IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 424–429.

Bowers, K. D. & et al. (2009). HAIL: A high-availability and integrity layer for cloud storage. *Proc. of the 16th ACM conference on Computer and communications security*, pp. 187–198.

Bravo-Marquez, F., Reeves, S. & Ugarte, M. (2019). Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions. *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pp. 119–124.

Bryden, T. S., Hilton, G., Dimitrov, B., de León, C. P. & Cruden, A. (2019). Rating a stationary energy storage system within a fast electric vehicle charging station considering user waiting times. *IEEE Transactions on Transportation Electrification*, 5(4), 879–889.

Bugday, A., Ozsoy, A. & Sever, H. (2019). Securing blockchain shards by using learning based reputation and verifiable random functions. *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–4.

Buşoniu, L., Babuška, R. & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1* (pp. 183–221). Springer.

Buterin, V., Conner, E., Dudley, R., Slipper, M., Norden, I. & Bakhta, A. (2019). EIP-1559: Fee market change for ETH 1.0 chain. *Published online*.

Cachin, C., Haas, R. & Vukolic, M. (2010). Dependable storage in the intercloud. *IBM research*, 3783, 1–6.

Cai, X., Geng, S., Zhang, J., Wu, D., Cui, Z., Zhang, W. & Chen, J. (2021). A Sharding scheme-based many-objective optimization algorithm for enhancing security in Blockchain-enabled industrial internet of things. *IEEE Transactions on Industrial Informatics*, 17(11), 7650–7658.

Cao, S., Miller, T., Foth, M., Powell, W., Boyen, X. & Turner-Morris, C. (2021). Integrating On-chain and Off-chain Governance for Supply Chain Transparency and Integrity. *arXiv preprint arXiv:2111.08455*.

Cao, Y., Tang, S., Li, C., Zhang, P., Tan, Y., Zhang, Z. & Li, J. (2011). An optimized EV charging model considering TOU price and SOC curve. *IEEE Transactions on Smart Grid*, 3(1), 388–393.

Cao, Y., Jiang, T., Kaiwartya, O., Sun, H., Zhou, H. & Wang, R. (2019). Toward pre-empted EV charging recommendation through V2V-based reservation system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(5), 3026–3039.

Carlsten, M., Kalodner, H., Weinberg, S. M. & Narayanan, A. (2016). On the instability of bitcoin without the block reward. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 154–167.

Castro, M., Liskov, B. et al. (1999). Practical byzantine fault tolerance. *OsDI*, 99(1999), 173–186.

Celik, Z. B., Babun, L., Sikder, A. K., Aksu, H., Tan, G., McDaniel, P. & Uluagac, A. S. (2018). Sensitive Information Tracking in Commodity {IoT}. *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1687–1704.

Chang, S.-Y., Park, Y., Wuthier, S. & Chen, C.-W. (2019). Uncle-block attack: Blockchain mining threat beyond block withholding for rational and uncooperative miners. *International Conference on Applied Cryptography and Network Security*, pp. 241–258.

Chason, E. D. (2019). Cryptocurrency Hard Forks and Revenue Ruling 2019-24. *Va. Tax Rev.*, 39, 279.

Chauhan, S. S., Pilli, E. S. & Joshi, R. C. (2021). BSS: a brokering model for service selection using integrated weighting approach in cloud environment. *Journal of Cloud Computing*, 10(1), 1–14.

Chen, J., Duan, K., Zhang, R., Zeng, L. & Wang, W. (2018a). An AI based super nodes selection algorithm in blockchain networks. *arXiv preprint arXiv:1808.00216*.

Chen, T., Zhang, B., Pourbabak, H., Kavousi-Fard, A. & Su, W. (2016). Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems. *IEEE Transactions on Smart Grid*, 9(4), 3563–3572.

Chen, W. & et al. (2019). Cooperative and distributed computation offloading for blockchain-empowered industrial Internet of Things. *IEEE Internet of Things Journal*, 6(5), 8433–8446.

Chen, X., Ji, J., Luo, C., Liao, W. & Li, P. (2018b). When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1178–1187.

Christidis, K. & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4, 2292–2303.

Chung, T., Nair, S., Ravi, U. & Kajgaonkar, P. (2021). Proof of Participation Voting for On-Chain Governance.

Clarke, R. & Wigan, M. (2011). You are where you've been: the privacy implications of location and tracking technologies. *Journal of Location Based Services*, 5(3-4), 138–155.

Coello, C. C. & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2, 1051–1056.

Conti, M., Kumar, E. S., Lal, C. & Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4), 3416–3452.

Courtois, N. T., Grajek, M. & Naik, R. (2013). The unreasonable fundamental incertitudes behind bitcoin mining. *arXiv preprint arXiv:1310.7935*.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G. et al. (2016). On scaling decentralized blockchains. *International conference on financial cryptography and data security*, pp. 106–125.

Dai, H.-N. & et al. (2019). Blockchain for Internet of Things: A survey. *IEEE Internet of Things Journal*, 6(5), 8076–8094.

Danish, S. M. (2019). A blockchain-based adaptive middleware for large scale internet of things data storage selection. *Proc. of Middleware'19 Doctoral Symposium*, pp. 17–19.

Danish, S. M. & et al. (2020). BlockAM: An Adaptive Middleware for Intelligent Data Storage Selection for Internet of Things. *2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pp. 61–71.

Danish, S. M., Lestas, M., Qureshi, H. K., Zhang, K., Asif, W. & Rajarajan, M. (2020a). Securing the LoRaWAN join procedure using blockchains. *Cluster Computing*, 23(3), 2123–2138.

Danish, S. M., Zhang, K. & Jacobsen, H.-A. (2020b). A Blockchain-Based Privacy-Preserving Intelligent Charging Station Selection for Electric Vehicles. *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–3.

Danish, S. M., Zhang, K., Jacobsen, H.-A., Ashraf, N. & Qureshi, H. K. (2020c). BlockEV: Efficient and Secure Charging Station Selection for Electric Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(7), 4194–4211.

De Filippi, P. & Loveluck, B. (2016). The invisible politics of bitcoin: governance crisis of a decentralized infrastructure. *Internet policy review*, 5(4).

Debe, M. & et al. (2019). IoT public fog nodes reputation system: A decentralized solution using Ethereum blockchain. *IEEE Access*, 7, 178082–178093.

Decker, C. & Wattenhofer, R. (2013). Information propagation in the bitcoin network. *IEEE P2P 2013 Proceedings*, pp. 1–10.

del Razo, V. & , H.-A. J. (2016). Smart charging schedules for highway travel with electric vehicles. *IEEE Transactions on Transportation Electrification*, 2(2), 160–173.

Dinh, T. N. & Thai, M. T. (2018). Ai and blockchain: A disruptive integration. *Computer*, 51(9), 48–53.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C. & Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. *Proceedings of the 2017 ACM international conference on management of data*, pp. 1085–1100.

Dobre, D. & et al. (2014). Hybris: Robust hybrid cloud storage. *Proc. of ACM Symposium on Cloud Computing*, pp. 1–14.

Dorri, A. & et al. (2017). Towards an optimized blockchain for IoT. *Proc. of the second IoTDI'17*, pp. 173–178.

Dziembowski, S., Faust, S., Kolmogorov, V. & Pietrzak, K. (2015). Proofs of space. *Annual Cryptology Conference*, pp. 585–605.

Eiza, M. H., Shi, Q., Marnerides, A. K., Owens, T. & Ni, Q. (2018). Efficient, Secure, and Privacy-Preserving PMIPv6 Protocol for V2G Networks. *IEEE Transactions on Vehicular Technology*, 68(1), 19–33.

Eskandarian, A., Chaoxian, W. & Chuanyang, S. (2019). Research Advances and Challenges of Autonomous and Connected Ground Vehicles. *IEEE Trans. on Intelligent Transportation Systems*, 22(2), 683–711.

Faria, C. & Correia, M. (2019). BlockSim: blockchain simulator. *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 439–446.

Ferreira, M. V., Moroz, D. J., Parkes, D. C. & Stern, M. (2021). Dynamic posted-price mechanisms for the blockchain transaction-fee market. *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pp. 86–99.

Fraiji, Y., Azzouz, L. B., Trojet, W. & Saidane, L. A. (2018). Cyber security issues of Internet of electric vehicles. *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6.

Fullmer, D. & Morse, A. S. (2018). Analysis of difficulty control in bitcoin and proof-of-work blockchains. *2018 IEEE Conference on Decision and Control (CDC)*, pp. 5988–5992.

G, B. & et al. (2019). Winning at the starting line: Joint network selection and service placement for mobile edge computing. *IEEE INFOCOM'19*, pp. 1459–1467.

Gai, K., Wu, Y., Zhu, L., Zhang, Z. & Qiu, M. (2019). Differential privacy-based blockchain for industrial internet-of-things. *IEEE Transactions on Industrial Informatics*, 16(6), 4156–4165.

Gao, F., Zhu, L., Shen, M., Sharif, K., Wan, Z. & Ren, K. (2018). A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks. *IEEE Network*, 32(6), 184–192.

Gao, Y., Kawai, S. & Nobuhara, H. (2019). Scalable blockchain protocol based on proof of stake and sharding. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 23(5), 856–863.

Georgiadis, E. (2019). How many transactions per second can bitcoin really handle? Theoretically. *Cryptology ePrint Archive*.

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H. & Capkun, S. (2016). On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 3–16.

Goli, M. & Eskandarian, A. (2014). A systematic multi-vehicle platooning and platoon merging: Strategy, control, and trajectory generation. *ASME 2014 Dynamic Systems and Control Conference*, 46193, V002T25A006.

Golosova, J. & Romanovs, A. (2018). Overview of the blockchain technology cases. *2018 59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, pp. 1–6.

Goyal, P., Sharma, A., Vyas, S. & Kumar, R. (2016). Customer and aggregator balanced dynamic Electric Vehicle charge scheduling in a smart grid framework. *IEEE ICEPES*, pp. 276–283.

Gueron, S., Johnson, S. & Walker, J. (2011). SHA-512/256. *2011 Eighth International Conference on Information Technology: New Generations*, pp. 354–358.

Guha, S. & Khuller, S. (1999). Greedy strikes back: Improved facility location algorithms. *Journal of algorithms*, 31(1), 228–248.

Gupta, S. S. (2017). Blockchain. *IBM Onlone (http://www. IBM. COM)*.

Gusrialdi, A., Qu, Z. & Simaan, M. A. (2017). Distributed scheduling and cooperative control for charging of electric vehicles at highway service stations. *IEEE Transactions on Intelligent Transportation Systems*, 18(10), 2713–2727.

Haouari, M. & Mhiri, M. (2021). A particle swarm optimization approach for predicting the number of COVID-19 deaths. *Scientific Reports*, 11(1), 1–13.

Haque, A. & Rahman, M. (2020). Blockchain technology: Methodology, application and security issues. *arXiv preprint arXiv:2012.13366*.

Harris, J. D. & Waggoner, B. (2019). Decentralized & collaborative AI on blockchain. *arXiv preprint arXiv:1907.07247*.

HC, T. & et al. (2009). *Introduction to Algorithms, Third Edition* (ed. 3rd). The MIT Press.

He, P., Tang, D. & Wang, J. (2020). Stake centralization in decentralized proof-of-stake Blockchain Network. *Available at SSRN*, 3609817.

He, T., Zhu, J., Zhang, J. & Zheng, L. (2018). An optimal charging/discharging strategy for smart electrical car parks. *Chinese Journal of Electrical Engineering*, 4(2), 28–35.

Hovland, G. & Kucera, J. (2017). Nonlinear feedback control and stability analysis of a proof-of-work blockchain.

Imai, S. & et al. (2018). A Performance Study of Geo-Distributed IoT Data Aggregation for Fog Computing. *2018 IEEE/ACM UCC Companion*, pp. 278–283.

Iwamura, M., Kitamura, Y., Matsumoto, T. & Saito, K. (2019). Can we stabilize the price of a cryptocurrency?: Understanding the design of Bitcoin and its potential to compete with Central Bank money. *Hitotsubashi Journal of Economics*, 41–60.

Jeong, S., Dao, N.-N., Lee, Y., Lee, C. & Cho, S. (2018). Blockchain Based Billing System for Electric Vehicle and Charging Station. *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 308–310.

Jiang, S., Zhang, X., Li, J., Yue, H. & Zhou, Y. (2020). Secure and privacy-preserving energy trading scheme based on blockchain. *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6.

Jiang, T., Fang, H. & Wang, H. (2018). Blockchain-based Internet of vehicles: distributed network architecture and performance analysis. *IEEE Internet of Things Journal*, 6(3), 4640–4649.

Jin, C., Tang, J. & Ghosh, P. (2013). Optimizing electric vehicle charging: A customer's perspective. *IEEE Transactions on Vehicular Technology*, 62(7), 2919–2927.

Jin, R., Zhang, X., Wang, Z., Sun, W., Yang, X. & Shi, Z. (2019). Blockchain-Enabled Charging Right Trading Among EV Charging Stations. *Energies*, 12(20), 3922.

John, T. & Pam, M. (2018). Complex adaptive blockchain governance. *MATEC Web of Conferences*, 223, 01010.

Junqueira, F. P., Reed, B. C. & Serafini, M. (2011). Zab: High-performance broadcast for primary-backup systems. *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, pp. 245–256.

Kamuni, V., Asfia, U., Sutavani, S., Sheikh, A. & Patel, D. (2019). Secure Energy Market against Cyber Attacks using Blockchain. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1792–1797.

Karlsson, K. & et al. (2018). Vegvisir: A partition-tolerant blockchain for the internet-of-things. *IEEE ICDCS'18*, pp. 1150–1158.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, 4, 1942–1948.

Khaled, A. E., Helal, A., Lindquist, W. & Lee, C. (2018). IoT-DDL–device description language for the "T" in IoT. *IEEE Access*, 6, 24048–24063.

Khodari, M., Rawat, A., Asplund, M. & Gurtov, A. (2019). Decentralized firmware attestation for in-vehicle networks. *5th on Cyber-Physical System Security Workshop*, pp. 47–56.

Kim, H., Park, J., Bennis, M. & Kim, S.-L. (2019). Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6), 1279–1283.

Knirsch, F., Unterweger, A. & Engel, D. (2018). Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. *Computer Science-Research and Development*, 33(1), 71–79.

Ko, R. K., Lee, B. S. & Pearson, S. (2011). Towards achieving accountability, auditability and trust in cloud computing. *International conference on advances in computing and communications*, pp. 432–444.

Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E. & Ford, B. (2018). Omniledger: A secure, scale-out, decentralized ledger via sharding. *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 583–598.

Konak, A., Coit, D. W. & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability engineering & system safety*, 91(9), 992–1007.

Kosba, A., Miller, A., Shi, E., Wen, Z. & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *2016 IEEE symposium on security and privacy (SP)*, pp. 839–858.

Kraft, D. (2016). Difficulty control for blockchain-based consensus systems. *Peer-to-peer Networking and Applications*, 9(2), 397–413.

Kumar, P. A. R. & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, 34(11), 1328–1341.

Kumar, R. S. & Saxena, A. (2011). Data integrity proofs in cloud storage. *IEEE Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pp. 1–4.

Kwon, Y., Kim, D., Son, Y., Vasserman, E. & Kim, Y. (2017). Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 195–209.

L, N. & et al. (2015). Safecoin: The decentralised network token. MaidSafe.

Larson, D. (2016). Distributed denial of service attacks–holding back the flood. *Network Security*, 2016(3), 5–7.

Lavi, R., Sattath, O. & Zohar, A. (2019). Redesigning Bitcoin's fee market. *The world wide web conference*, pp. 2950–2956.

Le, D. N., Le Tuan, L. & Tuan, M. N. D. (2019). Smart-building management system: An Internet-of-Things (IoT) application business model in Vietnam. *Technological Forecasting and Social Change*, 141, 22–35.

Lee, I. & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business horizons*, 58(4), 431–440.

Leonardos, S., Monnot, B., Reijsbergen, D., Skoulakis, E. & Piliouras, G. (2021). Dynamical analysis of the EIP-1559 Ethereum fee market. *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pp. 114–126.

Li, J., Zhou, Z., Wu, J., Li, J., Mumtaz, S., Lin, X., Gacanin, H. & Alotaibi, S. (2019). Decentralized on-demand energy supply for blockchain in internet of things: A microgrids approach. *IEEE Transactions on Computational Social Systems*, 6(6), 1395–1406.

Li, R., Song, T., Mei, B., Li, H., Cheng, X. & Sun, L. (2018). Blockchain for large-scale internet of things data storage and protection. *IEEE Transactions on Services Computing*, 12(5), 762–771.

Li, S., Yu, M., Yang, C.-S., Avestimehr, A. S., Kannan, S. & Viswanath, P. (2020a). Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously. *IEEE Transactions on Information Forensics and Security*, 16, 249–261.

Li, X., Jiang, P., Chen, T., Luo, X. & Wen, Q. (2020b). A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107, 841–853.

Li, Y. & Zhou, Y. (2021). Research on the Reciprocal Mechanism of Hybrid Governance in Blockchain. *Journal of Economics & Management Research. SRC/JESMR/127. DOI: doi. org/10.47363/JESMR/2021 (2)*, 121, 3.

Lin, F., Zheng, Z., Huang, Z., Tang, C., Peng, H. & Chen, Z. (2018). A sustainable reward mechanism for block mining in pow-based blockchain. *2018 5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, pp. 156–161.

Liu, B., Qin, Y. & Chu, X. (2019a). Reducing forks in the blockchain via probabilistic verification. *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, pp. 13–18.

Liu, D., Tan, K. C., Goh, C. K. & Ho, W. K. (2006). On solving multiobjective bin packing problems using particle swarm optimization. *2006 IEEE International Conference on Evolutionary Computation*, pp. 2095–2102.

Liu, W.-L., Gong, Y.-J., Chen, W.-N., Liu, Z., Wang, H. & Zhang, J. (2019b). Coordinated Charging Scheduling of Electric Vehicles: A Mixed-Variable Differential Evolution Approach. *IEEE Transactions on Intelligent Transportation Systems*, 21(12), 5094–5109.

Liu, Y., Qu, Y., Xu, C., Hao, Z. & Gu, B. (2021a). Blockchain-enabled asynchronous federated learning in edge computing. *Sensors*, 21(10), 3335.

Liu, Y., Lu, Q., Zhu, L., Paik, H.-Y. & Staples, M. (2021b). A systematic literature review on blockchain governance. *arXiv preprint arXiv:2105.05460*.

Liu, Y., Lu, Q., Yu, G., Paik, H.-Y. & Zhu, L. (2022). Defining blockchain governance principles: A comprehensive framework. *Information Systems*, 102090.

Long, Y., Chen, Y., Ren, W., Dou, H. & Xiong, N. N. (2020). Depet: a decentralized privacy-preserving energy trading scheme for vehicular energy network via blockchain and k-anonymity. *IEEE Access*, 8, 192587–192596.

Lu, X., Guan, Z., Zhou, X., Wu, L., Du, X. & Guizani, M. (2019). An efficient and privacy-preserving energy trading scheme based on blockchain. *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S. & Saxena, P. (2016). A secure sharding protocol for open blockchains. *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 17–30.

Ma, C., Li, J., Shi, L., Ding, M., Wang, T., Han, Z. & Poor, H. V. (2022). When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Computational Intelligence Magazine*, 17(3), 26–33.

Mahi, M., Baykan, Ö. K. & Kodaz, H. (2015). A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. *Applied Soft Computing*, 30, 484–490.

M*ακράκης*, G. M. (2018). Evaluating the performance of distributed Ledger-blockchain technologies in IoT environment.

Masood, A. B. & et al. (2019). Closing the Loop in Cyber-Physical Systems using Blockchain: Microgrid Frequency Control Example. *2019 2nd IEEE MENACOMM*, pp. 1–6.

Masood, A. B., Qureshi, H. K., Danish, S. M. & Lestas, M. (2019). Realizing an implementation platform for closed loop cyber-physical systems using blockchain. *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5.

McGinn, D., Birch, D., Akroyd, D., Molina-Solana, M., Guo, Y. & Knottenbelt, W. J. (2016). Visualizing dynamic bitcoin transaction patterns. *Big data*, 4(2), 109–119.

Memon, R. A., Li, J. P. & Ahmed, J. (2019). Simulation model for blockchain systems using queuing theory. *Electronics*, 8(2), 234.

Miller, A., Litton, J., Pachulski, A., Gupta, N., Levin, D., Spring, N. & Bhattacharjee, B. (2015). Discovering bitcoin's public topology and influential nodes. *et al*.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W. & Qijun, C. (2017). A review on consensus algorithm of blockchain. *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2567–2572.

Mišić, V. B., Mišić, J. & Chang, X. (2019). On forks and fork characteristics in a Bitcoin-like distribution network. *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 212–219.

Miyachi, K. & Mackey, T. K. (2021). hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design. *Information Processing & Management*, 58(3), 102535.

Moghaddam, M. & Davis, J. G. (2014). Service selection in web service composition: A comparative review of existing approaches. *Web services foundations*, 321–346.

Mohammed, A. H., Abdulateef, A. A. & Abdulateef, I. A. (2021). Hyperledger, Ethereum and Blockchain Technology: A Short Overview. *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6.

Mollah, M. & et al. (2017). Secure data sharing and searching at the edge of cloud-assisted internet of things. *IEEE Cloud Computing*, 34–42.

Mukherjee, J. C. & Gupta, A. (2014). A mobility aware scheduler for low cost charging of electric vehicles in smart grid. *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–8.

Mwale, M. (2016). *Modelling the dynamics of the bitcoin blockchain*. (Ph.D. thesis, Stellenbosch: Stellenbosch University).

Nabilou, H. (2020). Bitcoin governance as a decentralized financial market infrastructure. *Stan. J. Blockchain L. & Pol'y*, 4, 1.

Nakamoto, S. (2008). Re: Bitcoin P2P e-cash paper. *The Cryptography Mailing List*.

Nakamoto, S. et al. (2008). Bitcoin: A peer-to-peer electronic cash system.(2008).

Narayanan, A., Bonneau, J., Felten, E., Miller, A. & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press.

Nartey, C., Tchao, E. T., Gadze, J. D., Yeboah-Akowuah, B., Nunoo-Mensah, H., Welte, D. & Sikora, A. (2022). Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2022(1), 1–27.

Nasir, Q., Qasse, I. A., Talib, M. A. & Nassif, A. B. (2018). Performance Analysis of Hyperledger Fabric Platforms. *Security and Communication Networks*, 2018, 1–14. doi: 10.1155/2018/3976093.

Nejad, M. M., Mashayekhy, L., Chinnam, R. B. & Grosu, D. (2017). Online scheduling and pricing for electric vehicle charging. *IISE Transactions*, 49(2), 178–193.

Nguyen, L. N., Nguyen, T. D., Dinh, T. N. & Thai, M. T. (2019). Optchain: optimal transactions placement for scalable blockchain sharding. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 525–535.

Nimalsiri, N. I., Mediwaththe, C. P., Ratnam, E. L., Shaw, M., Smith, D. B. & Halgamuge, S. K. (2019). A survey of algorithms for distributed charging control of electric vehicles in smart grid. *IEEE Transactions on Intelligent Transportation Systems*, 21(11), 4497–4515.

Nofer, M., Gomber, P., Hinz, O. & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3), 183–187.

Nord, J. H., Koohang, A. & Paliszkiewicz, J. (2019). The Internet of Things: Review and theoretical framework. *Expert Systems with Applications*, 133, 97–108.

Ongaro, D. & Ousterhout, J. (2014). In search of an understandable consensus algorithm. *2014 USENIX Annual Technical Conference (Usenix ATC 14)*, pp. 305–319.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *International conference on the theory and applications of cryptographic techniques*, pp. 223–238.

Pajic, J., Rivera, J., Zhang, K. & Jacobsen, H.-A. (2018). Eva: Fair and auditable electric vehicle charging service using blockchain. *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pp. 262–265.

Pandey, S., Ojha, G., Shrestha, B. & Kumar, R. (2019). BlockSIM: A practical simulation tool for optimal network design, stability and planning. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 133–137.

Papaioannou, T. G., Bonvin, N. & Aberer, K. (2012). Scalia: An adaptive scheme for efficient multi-cloud storage. *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–10.

Park, S., Kwon, A., Fuchsbauer, G., Gaži, P., Alwen, J. & Pietrzak, K. (2018). Spacemint: A cryptocurrency based on proofs of space. *International Conference on Financial Cryptography and Data Security*, pp. 480–499.

Passerat-Palmbach, J., Farnan, T., Miller, R., Gross, M. S., Flannery, H. L. & Gleim, B. (2019). A blockchain-orchestrated federated learning architecture for healthcare consortia. *arXiv preprint arXiv:1910.12603*.

Patil, H. & Kalkhambkar, V. N. (2019). Charging cost minimisation by centralised controlled charging of electric vehicles. *International Transactions on Electrical Energy Systems*, 30(2), e12226.

Pervaiz, S., Ul-Qayyum, Z., Bangyal, W. H., Gao, L. & Ahmad, J. (2021). A systematic literature review on particle swarm optimization techniques for medical diseases detection. *Computational and Mathematical Methods in Medicine*, 2021.

Popper, N. (2013). In Bitcoin's orbit: Rival virtual currencies vie for acceptance. *New York Times*, 4.

Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 63(3), 337–370.

Pourazarm, S., Cassandras, C. G. & Malikopoulos, A. (2014). Optimal routing of electric vehicles in networks with charging nodes: A dynamic programming approach. *2014 IEEE International Electric Vehicle Conference (IEVC)*, pp. 1–7.

Psaras, Y. & Dias, D. (2020). The interplanetary file system and the filecoin network. *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pp. 80–80.

Pustišek, M., Kos, A. & Sedlar, U. (2016). Blockchain based autonomous selection of electric vehicle charging station. *2016 international conference on identification, information and knowledge in the Internet of Things (IIKI)*, pp. 217–222.

Pyrkova, A. & Temirbekova, Z. (2020). Compare encryption performance across devices to ensure the security of the IOT. *Indonesian Journal of Electrical Engineering and Computer Science*, 20(2), 894–902.

Qin, H. & Zhang, W. (2011). Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*, pp. 51–60.

Qiu, C., Yu, F. R., Yao, H., Jiang, C., Xu, F. & Zhao, C. (2018). Blockchain-based software-defined industrial Internet of Things: A dueling deep ${Q} $-learning approach. *IEEE Internet of Things Journal*, 6(3), 4627–4639.

Qiu, C., Ren, X., Cao, Y. & Mai, T. (2020). Deep reinforcement learning empowered adaptivity for future blockchain networks. *IEEE Open Journal of the Computer Society*, 2, 99–105.

Radi, E. M., Lasla, N., Bakiras, S. & Mahmoud, M. (2019). Privacy-Preserving Electric Vehicle Charging for Peer-to-Peer Energy Trading Ecosystems. *IEEE International Conference on Communications (ICC)*, pp. 1–6.

Rafique, A. & et al. (2017). Towards an adaptive middleware for efficient multi-cloud data storage. *CrossCloud'17*, pp. 4.

Rafique, A. & et al. (2019). SCOPE: self-adaptive and policy-based data management middleware for federated clouds. *Journal of Internet Services and Applications*, 10(1), 2.

Rahman, M. S., Khalil, I., Alabdulatif, A. & Yi, X. (2019). Privacy preserving service selection using fully homomorphic encryption scheme on untrusted cloud service platform. *Knowledge-Based Systems*, 180, 104–115.

Ramanan, P., Nakayama, K. & Sharma, R. (2019). BAFFLE: Blockchain based aggregator free federated learning. *arXiv preprint arXiv:1909.07452*.

Ramnath, S., Javali, A., Narang, B., Mishra, P. & Routray, S. K. (2017). IoT based localization and tracking. *2017 International Conference on IoT and Application (ICIOT)*, pp. 1–4.

Rao, R. V., Rai, D., Ramkumar, J. & Balic, J. (2016). A new multi-objective Jaya algorithm for optimization of modern machining processes. *Advances in Production Engineering & Management*, 11(4).

Rathee, G., Sharma, A., Iqbal, R., Aloqaily, M., Jaglan, N. & Kumar, R. (2019). A blockchain framework for securing connected and autonomous vehicles. *Sensors*, 19(14), 3165.

Reijsbergen, D., Sridhar, S., Monnot, B., Leonardos, S., Skoulakis, S. & Piliouras, G. (2021). Transaction Fees on a Honeymoon: Ethereum's EIP-1559 One Month Later. *2021 IEEE International Conference on Blockchain (Blockchain)*, pp. 196–204.

Ridha, H. M., Gomes, C., Hizam, H., Ahmadipour, M., Heidari, A. A. & Chen, H. (2021). Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 135, 110202.

Ritz, F. & Zugenmaier, A. (2018). The impact of uncle rewards on selfish mining in ethereum. *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 50–57.

Rivera, J., Goebel, C. & Jacobsen, H.-A. (2016). Distributed convex optimization for electric vehicle aggregators. *IEEE Transactions on Smart Grid*, 8(4), 1852–1863.

Rivest, R. L., Shamir, A. & Tauman, Y. (2001). How to leak a secret. *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565.

Rosa, E., D'Angelo, G. & Ferretti, S. (2019). Agent-based simulation of blockchains. *Asian Simulation Conference*, pp. 115–126.

Roughgarden, T. (2020). Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. *arXiv preprint arXiv:2012.00854*.

Roughgarden, T. (2021). Transaction fee mechanism design. *ACM SIGecom Exchanges*, 19(1), 52–55.

Rubio, J. E., Alcaraz, C. & Lopez, J. (2018). Addressing Security in OCPP: Protection Against Man-in-the-Middle Attacks. *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5.

Ruffing, T., Kate, A. & Schröder, D. (2015). Liar, liar, coins on fire! Penalizing equivocation by loss of bitcoins. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 219–230.

Ruiz-Alvarez, A. & Humphrey, M. (2011). An automated approach to cloud storage service selection. *Proceedings of the 2nd international workshop on Scientific cloud computing*, pp. 39–48.

Saad, S. M. S., Radzi, R. Z. R. M. & Othman, S. H. (2021). Comparative Analysis of the Blockchain Consensus Algorithm Between Proof of Stake and Delegated Proof of Stake. *2021 International Conference on Data Science and Its Applications (ICoDSA)*, pp. 175–180.

Saghezchi, F. B., Mantas, G., Ribeiro, J., Al-Rawi, M., Mumtaz, S. & Rodriguez, J. (2017). Towards a secure network architecture for smart grids in 5G era. *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 121–126.

Said, D., Cherkaoui, S. & Khoukhi, L. (2013). Queuing model for EVs charging at public supply stations. *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 65–70.

Saied, A., Overill, R. E. & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*, 172, 385–393.

Saito, K. & Iwamura, M. (2019). How to make a digital currency on a blockchain stable. *Future Generation Computer Systems*, 100, 58–69.

Salimitari, M., Joneidi, M. & Chatterjee, M. (2019). AI-enabled Blockchain: An Outlier-aware Consensus Protocol for Blockchain-based IoT Networks. *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

Samuel, O., Javaid, N., Shehzad, F., Iftikhar, M. S., Iftikhar, M. Z., Farooq, H. & Ramzan, M. (2019). Electric Vehicles Privacy Preserving Using Blockchain in Smart Community. *International Conference on Broadband and Wireless Computing, Communication and Applications*, pp. 67–80.

Samy, A., Yu, H., Zhang, H. & Zhang, G. (2021). SPETS: Secure and Privacy-Preserving Energy Trading System in Microgrid. *Sensors*, 21(23), 8121.

Scherer, M. (2017). Performance and scalability of blockchain networks and smart contracts.

Schmidt, K., Saucke, F. & Spengler, T. S. (2018). Scheduling of Electric Vehicles in the Police Fleet. In *Operations Research Proceedings 2017* (pp. 693–699). Springer.

Seike, H., Aoki, Y. & Koshizuka, N. (2019). Fork rate-based analysis of the longest chain growth time interval of a pow blockchain. *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 253–260.

Shafagh, H. & et al. (2017). Towards blockchain-based auditable storage and sharing of iot data. *Proceedings of the 2017 on Cloud Computing Security Workshop*, pp. 45–50.

Shahsavari, Y., Zhang, K. & Talhi, C. (2019a). Performance modeling and analysis of the bitcoin inventory protocol. *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pp. 79–88.

Shahsavari, Y., Zhang, K. & Talhi, C. (2019b). A theoretical model for fork analysis in the bitcoin network. *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 237–244.

Shi, W. & Wong, V. W. (2011). Real-time vehicle-to-grid control algorithm under price uncertainty. *IEEE SmartGridComm*, pp. 261–266.

Shibata, N. (2019a). Blockchain Consensus Formation while Solving Optimization Problems. *arXiv preprint arXiv:1908.01915*.

Shibata, N. (2019b). Proof-of-search: combining blockchain consensus formation with solving optimization problems. *IEEE Access*, 7, 172994–173006.

Singh, N. & Vardhan, M. (2020). Computing optimal block size for blockchain based applications with contradictory objectives. *Procedia Computer Science*, 171, 1389–1398.

Singh, N. & Vardhan, M. (2021). Multi-objective optimization of block size based on CPU power and network bandwidth for blockchain applications. *Proceedings of the fourth international conference on microelectronics, computing and communication systems*, pp. 69–78.

Sirer, E. & Eyal, I. (2014). How to Disincentivize Large Bitcoin Mining Pools. *Blog post: http://hackingdistributed. com/2014/06/18/how-to-disincentivize-large-bitcoin-miningpools*.

Solat, S. & Potop-Butucaru, M. (2016). Zeroblock: Preventing selfish mining in bitcoin. *arXiv preprint arXiv:1605.02435*.

Son, Y.-B., Im, J.-H., Kwon, H.-Y., Jeon, S.-Y. & Lee, M.-K. (2020). Privacy-preserving peer-to-peer energy trading in blockchain-enabled smart grids using functional encryption. *Energies*, 13(6), 1321.

Spasovski, J. & Eklund, P. (2017). Proof of stake blockchain: performance and scalability for groupware communications. *Proceedings of the 9th International Conference on Management of Digital EcoSystems*, pp. 251–258.

Squicciarini, A., Carminati, B. & Karumanchi, S. (2011). A privacy-preserving approach for web service selection and provisioning. *2011 IEEE International Conference on Web Services*, pp. 33–40.

Stoykov, L., Zhang, K. & Jacobsen, H.-A. (2017). Vibes: fast blockchain simulations for large-scale peer-to-peer networks. *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos*, pp. 19–20.

Su, Z., Wang, Y., Xu, Q., Fei, M., Tian, Y.-C. & Zhang, N. (2018). A secure charging scheme for electric vehicles with smart communities in energy blockchain. *IEEE Internet of Things Journal*, 6(3), 4601–4613.

Sun, Y., Yin, L., Sun, Z., Tian, Z. & Du, X. (2020). An IoT data sharing privacy preserving scheme. *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 984–990.

Sunyaev, A. (2020). Distributed ledger technology. In *Internet Computing* (pp. 265–299). Springer.

Sweda, T. M. & Klabjan, D. (2012). Finding minimum-cost paths for electric vehicles. *IEEE International Electric Vehicle Conference*, pp. 1–4.

Syafruddin, W. A., Dadkhah, S. & Köppen, M. (2019). Blockchain Scheme Based on Evolutionary Proof of Work. *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 771–776.

Syed, T. A., Alzahrani, A., Jan, S., Siddiqui, M. S., Nadeem, A. & Alghamdi, T. (2019). A comparative analysis of blockchain architecture and its applications: Problems and recommendations. *IEEE access*, 7, 176838–176869.

Tao, Y., Li, B., Jiang, J., Ng, H. C., Wang, C. & Li, B. (2020). On sharding open blockchains with smart contracts. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 1357–1368.

Thakkar, A. & Chaudhari, K. (2021). A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. *Archives of Computational Methods in Engineering*, 28(4), 2133–2164.

Tsaousoglou, G., Steriotis, K. & Varvarigos, E. (2019). A stochastic approximation method for price-based assignment of Electric Vehicles to Charging Stations. *IEEE International Conference on Smart Energy Systems and Technologies (SEST)*, pp. 1–6.

Tuan, M. N. D., Thanh, N. N. & Le Tuan, L. (2019). Applying a mindfulness-based reliability strategy to the Internet of Things in healthcare–A business model in the Vietnamese market. *Technological Forecasting and Social Change*, 140, 54–68.

Vaquero, L. M. & Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, 44(5), 27–32.

Wang, J. & Wang, H. (2019). Monoxide: Scale out blockchains with asynchronous consensus zones. *16th USENIX symposium on networked systems design and implementation (NSDI 19)*, pp. 95–112.

Wang, Q. & et al. (2009). Enabling public verifiability and data dynamics for storage security in cloud computing. *Proc. of ESORICS'09,*, pp. 355–370.

Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y. & Kim, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7, 22328–22370.

Watkins, C. J. & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4), 279–292.

Wen-Bin, Y. & Yong-Hong, D. (2021). W-MOPSO in adaptive circuits for blast wave measurements. *IEEE Sensors Journal*, 21(7), 9323–9332.

Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014), 1–32.

Wu, Z. & et al. (2013). Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services. *Proc. of ACM Symposium on Operating Systems Principles*, pp. 292–308.

Xia, Y., Chen, P., Bao, L., Wang, M. & Yang, J. (2011). A QoS-aware web service selection algorithm based on clustering. *2011 IEEE International Conference on Web Services*, pp. 428–435.

Xu, S., Chen, X. & He, Y. (2021). EVchain: An Anonymous Blockchain-Based System for Charging-Connected Electric Vehicles. *Tsinghua Science and Technology*, 26(6), 845–856.

Yang, J. & Yang, Z. (2014). Pricing scheme for aggregate load scheduling of plug-in electric taxi fleet. *Proceedings of the 33rd Chinese Control Conference*, pp. 7589–7594.

Yang, R., Yu, F. R., Si, P., Yang, Z. & Zhang, Y. (2019). Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(2), 1508–1532.

Yang, S.-N., Cheng, W.-S., Hsu, Y.-C., Gan, C.-H. & Lin, Y.-B. (2013). Charge scheduling of electric vehicles in highways. *Mathematical and Computer Modelling*, 57(11-12), 2873–2882.

Yang, W., Guan, Z., Wu, L., Du, X., Lv, Z. & Guizani, M. (2020). Autonomous and Privacy-preserving Energy Trading Based on Redactable Blockchain in Smart Grid. *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6.

Yang, Y., Liu, J. & Tan, S. (2021). A multi-objective evolutionary algorithm for steady-state constrained multi-objective optimization problems. *Applied Soft Computing*, 101, 107042.

Yasaweerasinghelage, R., Staples, M. & Weber, I. (2017). Predicting latency of blockchain-based systems using architectural modelling and simulation. *2017 IEEE International Conference on Software Architecture (ICSA)*, pp. 253–256.

Yi, H., Guan, K., He, D., Ai, B., Dou, J. & Kim, J. (2019). Characterization for the Vehicle-to-Infrastructure Channel in Urban and Highway Scenarios at the Terahertz Band. *IEEE Access*, 7, 166984–166996.

Yoo, H., Yim, J. & Kim, S. (2018). The blockchain for domain based static sharding. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 1689–1692.

Yu, F. R., Liu, J., He, Y., Si, P. & Zhang, Y. (2018). Virtualization for distributed ledger technology (vDLT). *IEEE Access*, 6, 25019–25028.

Yu, Y., Song, T., Su, C., Tang, X. & Han, Z. (2019). Hierarchical Game for Electric Vehicle Public Charging Market. *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6.

Yuan, S., Cao, B., Peng, M. & Sun, Y. (2021). ChainsFL: Blockchain-driven Federated Learning from Design to Realization. *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6.

Yun, J., Goh, Y. & Chung, J.-M. (2019). Trust-based shard distribution scheme for fault-tolerant shard blockchain networks. *IEEE Access*, 7, 135164–135175.

Yun, J., Goh, Y. & Chung, J.-M. (2020). DQN-based optimization framework for secure sharded blockchain systems. *IEEE Internet of Things Journal*, 8(2), 708–722.

Z, J. & et al. (2019). Data Centers Selection for Moving Geo-distributed Big Data to Cloud. *Journal of Internet Technology*, 20(1), 111–122.

Zamani, M., Movahedi, M. & Raykova, M. (2018). Rapidchain: Scaling blockchain via full sharding. *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 931–948.

Zargham, M., Zhang, Z. & Preciado, V. (2018). A state-space modeling framework for engineering blockchain-enabled economic systems. *arXiv preprint arXiv:1807.00955*.

Zhang, D., Le, J., Lei, X., Xiang, T. & Liao, X. (2021a). Exploring the redaction mechanisms of mutable blockchains: A comprehensive survey. *International Journal of Intelligent Systems*, 36(9), 5051–5084.

Zhang, J., Sheng, J., Lu, J. & Shen, L. (2021b). UCPSO: a uniform initialized particle swarm optimization algorithm with cosine inertia weight. *Computational Intelligence and Neuroscience*, 2021.

Zhang, K. & et al. (2018). Deconstructing Blockchains: Concepts, Systems, and Insights. *DEBS*, pp. 187–190.

Zhang, P., Wang, L., Li, C. & Zhou, M. (2020). An Optimization Model for Transaction Placement in Blockchain Shards. *IFAC-PapersOnLine*, 53(5), 374–378.

Zhang, P., Schmidt, D. C., White, J. & Dubey, A. (2019a). Consensus mechanisms and information security technologies. *Advances in Computers*, 115, 181–209.

Zhang, Y., Xu, C., Lin, X. & Shen, X. S. (2019b). Blockchain-based public integrity verification for cloud storage against procrastinating auditors. *IEEE Transactions on Cloud Computing*, 9(3), 923–937.

Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D., Li, Z., Lyu, L. & Liu, Y. (2020). Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet of Things Journal*, 8(3), 1817–1829.

Zheng, Z., Xie, S., Dai, H., Chen, X. & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *2017 IEEE international congress on big data (BigData congress)*, pp. 557–564.

Zhou, Z., Xiong, F., Xu, C., He, Y. & Mumtaz, S. (2017). Energy-efficient vehicular heterogeneous networks for green cities. *IEEE Transactions on industrial Informatics*, 14(4), 1522–1531.