Journal of
Sensor and Actuator Networks

MDPI

Article

Game Theory-Based Incentive Design for Mitigating Malicious Behavior in Blockchain Networks

Souhail Mssassi * and Anas Abou El Kalam

National School of Applied Sciences, Cadi Ayyad University, Marrakech 40000, Morocco; a.abouelkalam@uca.ac.ma
* Correspondence: souhail.mssassi@owasp.org

Abstract: This paper presents an innovative incentive model that utilizes graph and game theories to address the issue of node incentives in decentralized blockchain networks such as EVM blockchains. The lack of incentives for nodes within EVM networks gives rise to potential weaknesses that might be used for various purposes, such as broadcasting fake transactions or withholding blocks. This affects the overall trust and integrity of the network. To address this issue, the current study offers a network model that incorporates the concepts of graph theory and utilizes a matrix representation for reward and trust optimization. Furthermore, this study presents a game-theoretic framework that encourages cooperative conduct and discourages malicious actions, ultimately producing a state of equilibrium according to the Nash equilibrium. The simulations validated the model's efficacy in addressing fraudulent transactions and emphasized its scalability, security, and fairness benefits. This study makes a valuable contribution to the field of blockchain technology by presenting an incentive model that effectively encourages the development of secure and trusted decentralized systems.

Keywords: incentive model; decentralized blockchain networks; node incentives; graph theory; game theory; EVM networks; reward optimization; Nash's equilibrium

Citation: Mssassi, S.; Abou El Kalam, A. Game Theory-Based Incentive Design for Mitigating Malicious Behavior in Blockchain Networks. *J. Sens. Actuator Netw.* **2024**, *13*, 7. https://doi.org/10.3390/jsan13010007

Academic Editors: Ki-Hyun Jung, Luis Javier García Villalba and Lei Shu

Received: 20 November 2023
Revised: 8 January 2024
Accepted: 9 January 2024
Published: 15 January 2024

1. Introduction

Blockchain technology has permeated various sectors [1] and has been lauded for its potential to instill immutability, transparency, and decentralization in systems, thereby revolutionizing traditional systems. At the core of these decentralized networks, nodes are imperative for executing and validating transactions, thereby safeguarding the integrity of the blockchain. Within this ecosystem, nodes, particularly those executing transactions encompassing native currency transfers and function executions, are pivotal yet often lack adequate incentives, in contrast to mining nodes.

The motivation for this research stems from a critical gap in current blockchain models—the need for more incentivization of nodes, especially non-mining ones, within EVM blockchains. This disparity not only compromises network security and efficiency but also poses a significant threat to the integrity and robustness of the blockchain. Our objective is to develop an incentive model that ensures node cooperation and network integrity and addresses scalability and security challenges, thereby reinforcing the overall trust within the network.

Despite extensive research on blockchain technology, a notable gap still exists in understanding how to effectively incentivize nodes to align individual gains with the network's overall health. This study seeks to address this gap by asking the following research questions:

1. How can we integrate graph and game theories to create a robust incentive model for nodes within EVM blockchains?
2. How can a trust matrix influence node behavior to enhance network security and efficiency?
3. How does the proposed model ensure fairness and scalability and discourage fraudulent activities within blockchain networks?

To elaborate on this approach, imagine a scenario in which individual nodes driven by self-interest might prioritize transactions that offer higher financial incentives, which aligns with the principles encapsulated in the concept of maximum extractable value (MEV) [2]. However, by synergizing this self-interest with group interests, the framework ensures that the nodes consider the overall health and security of the network when making decisions. This balance ensures that, while nodes act to maximize their gains, they do not compromise the interests of the network, leading to a more resilient and robust blockchain system.

Therefore, our research endeavors to develop an intricately designed framework to navigate this problem. The framework posits a dual-faceted approach: incentivizing nodes via financial gains from honest actions and broadcasting to numerous nodes while enabling decision making based on node trust. The "trust matrix" is central to this approach. It is a novel construct wherein nodes store and dynamically adjust trust coefficients based on peer actions, thereby perpetually recalibrating their interactions and decisions within the network.

This matrix functions using the algorithms discussed in further chapters, which adjust coefficients based on observed behaviors, reinforce honest actions, and penalize malicious activities. This continuous recalibration of trust coefficients serves as a feedback mechanism, ensuring that the nodes remain aligned with the broader objectives of the network. If a node behaves in a way that benefits solely itself but is detrimental to the group, its trust coefficient may decrease, potentially leading to disadvantages in future interactions. Consequently, the framework naturally encourages actions advantageous to the individual node and the group. This leads to a state of Pareto Optimality [3], in which no node's situation can improve without adversely affecting another, thereby achieving an efficient balance between individual and collective benefits.

The methodology adopted in this study involved three pivotal steps. First, a model of the blockchain network was crafted using graph theory, representing each executor as a node within the graph. Subsequently, each node maintains a trust matrix, where the trust coefficients for the peer nodes are stored and adjusted based on their actions. Finally, nodes seek to optimize their gains via continuous adjustments and modifications within this matrix, ensuring alignment with individual and group interests. The specific algorithms and mechanisms employed in this process are described in detail in the following sections.

This paper is structured as follows: Section 2 elucidates fundamental concepts in blockchain and game theory, laying a foundation for comprehending the ensuing framework. Section 3 explores the challenges of incentivizing nodes and examines related studies in this domain. Section 4 describes the modeling of the blockchain network using a probabilistic matrix and defines our game characteristics; Section 5 unveils our framework, discussing node communication and trust matrix updates. Section 6 articulates the reward system, elucidates how nodes are rewarded based on their actions, and subsequently explores the various actions a node can undertake to optimize its gain. Section 7 delves into how nodes, driven by maximizing rewards, engage in strategic actions influenced by a trust matrix. It discusses the optimization problem nodes face to balance individual gains with collective network benefits. Section 8 provides a detailed overview of our comprehensive simulations, highlighting our model's network typologies and scalability diversity. It elaborates on how the simulations demonstrate the framework's efficacy in various network environments. Finally, Section 9 describes the essence of our research. This section summarizes the contributions of our study to the blockchain field, reflecting the implications of our innovative incentive model for network integrity and participation. It also outlines potential future research directions, emphasizing blockchain technology's continuous evolution and dynamic nature.

2. Background

The following section examines the blockchain network's primary constituents and game theory concepts. While numerous other components cannot be elaborated upon

within the scope of this discussion, the ensuing components are primarily associated with the dissemination of transactions and the incentivization problem among nodes.

2.1. Blocks

Blocks are fundamental components of a blockchain database and serve as data structures that permanently store transaction data in a blockchain [4]. A block stores a subset of all the latest transactions the network still needs to validate. After validating the data, the block is stored in a ledger. Subsequently, a new block is generated to accommodate the inclusion and verification of recent transactions. A block is a repository of data that, once written, is immutable and cannot be modified or deleted. Figure 1 illustrates the general format of chained blocks, highlighting the following elements of information within a block.

- The term "magic number" refers to a numerical figure encompassing distinct characteristics and serving as an identifier for a single block inside the network.
- The block size parameter establishes a predetermined restriction on the size of a block, which restricts the amount of data that may be inputted.
- The block header is a component that contains pertinent information on the block.
- The transaction counter is a numerical value that indicates the total number of transactions in a given block.
- Transactions refer to a comprehensive batch of all recorded transactions in a particular block [4].

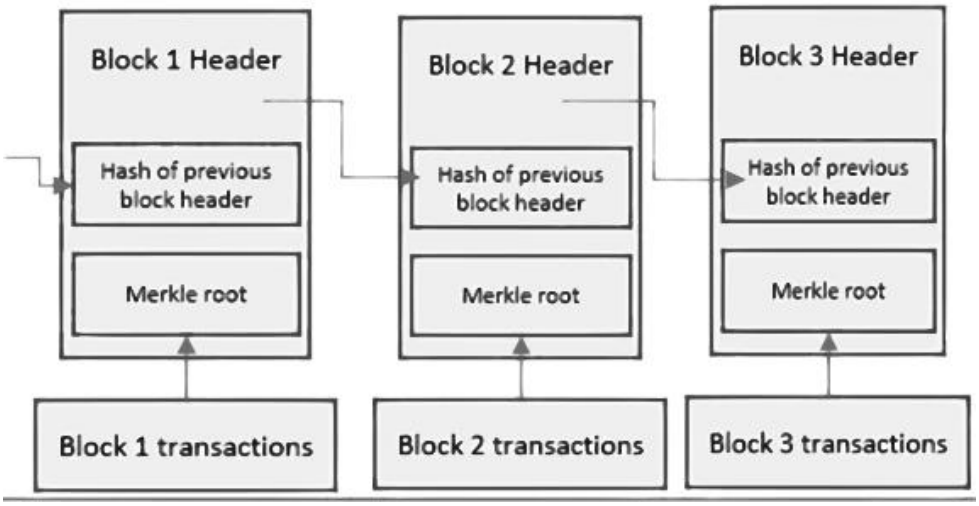

Figure 1. General format of chained blocks.

2.2. Transactions

Blockchain transactions refer to transmitting data via a network of computers within a blockchain system. In a blockchain, interconnected nodes form a network that collectively maintains transactional data as replicated copies, often known as a digital ledger [5]. Figure 2 presents an example of a Bitcoin transaction, illustrating how data transmission is facilitated within such a network.

2.3. Nodes

Blockchain nodes are integral components within a network, authorized to maintain a distributed ledger and act as communication hubs for various network responsibilities. While sharing foundational purposes, these nodes can be categorized into specific roles based on their functionalities.

One of the critical categories is the transaction-executing nodes. These nodes are primarily tasked with the validation and verification of the transactions. The core role of a blockchain node within this category involves ensuring the legitimacy of each sequential

set of transactions, known as blocks. Among these, full nodes represent a significant subset. They retain and record all transactions within the blockchain in their storage and perform the critical function of validating blocks and transactions. In contrast, lightweight nodes, another form of transaction-executing node, have reduced storage requirements. They primarily focus on retrieving block headers for transaction verification and minimizing their data storage requirements.

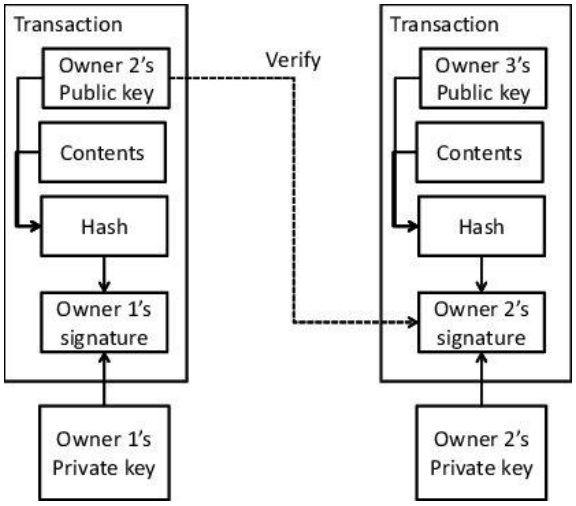

Figure 2. Example of a Bitcoin transaction.

Another pivotal category is the mining nodes. These nodes engage in the generation and integration of new blocks into the blockchain. The process begins when a miner attempts to add a new block of transactions to the blockchain and broadcasts this information across the network. It is important to note that the allocation of a block reward varies and is not a consistent feature of any blockchain network. Table 1 presents a comparison between transaction-executing nodes and mining nodes, highlighting the distinct roles and functionalities of these two key categories within a blockchain network.

Table 1. Comparison: transaction-executing nodes vs. mining nodes.

Aspect	Transaction-Executing Nodes	Mining Nodes
Purpose	Validate and relay transactions across the network.	Add new blocks to the blockchain.
Primary Responsibility	Ensure transactions comply with network rules.	Collect, verify, and process transactions into blocks.
Reward System	Do not typically receive cryptocurrency rewards.	Receive rewards in the form of minted cryptocurrency.
Hardware Requirements	Do not require significant computational power.	Requires high computational power.
Storage	It may or may not store the entire blockchain history.	Store blockchain as the whole history and validation transactions.
Incentive	Maintain network integrity for their applications.	Monetary rewards for adding successful blocks.

Nodes have the discretion to accept or reject a block based on their authenticity, which is determined by the validity of their signature and transaction [6]. When a node receives a new block of transactions, it saves and stores its highest position within the current chain of blocks. In summary, nodes perform the following functions.

1. Nodes play a crucial role in the validation process of determining the legitimacy of a block of transactions because they have the authority to accept or reject it.

2. The nodes are responsible for preserving and retaining transaction blocks, including the entire transaction record in the blockchain.
3. The transaction history is distributed and propagated by nodes to other nodes, which may require synchronization with the blockchain [4].

It is important to note that our research specifically addresses the challenges and dynamics associated with executor nodes within a blockchain network. In particular, executor nodes execute transaction requests and append new blocks to the blockchain. Focusing on these nodes, we aim to solve distinct issues pertinent to their operations, such as incentivization, trust management, and decision-making processes in validating and storing transactions. This specification narrows the scope of our study, allowing for a more in-depth and targeted exploration of solutions that enhance the functionality and reliability of executor nodes in decentralized blockchain systems.

2.4. Game Theory

Game theory, a pivotal theoretical framework in disciplines such as economics and computer science, rigorously examines interactions characterized by formalized incentives and outcomes [7]. This mathematical framework facilitates the analytical exploration of strategic interactions among rational entities, particularly in scenarios in which an agent's payoff depends on the actions undertaken by other agents [8].

Strategic decision making is especially salient within the blockchain domain. Blockchain networks can be conceptualized as intricate ecosystems of nodes (or agents) continuously engaging in decisions, such as validating a transaction or selecting transactions for block inclusion. These decisions are not autonomous but interdependent and influenced by other network participants' potential actions and determinations. In this context, game theory provides the tools for modeling these multifaceted decisions, encompassing all network entities' potential strategies and actions [9]. This is crucial for anticipating node behavior when confronted with choices that can invariably impact rewards or overarching network security.

In a blockchain ecosystem, the application of game theory is invaluable for modeling and analyzing the strategic interactions of nodes. This includes decision making, strategies, and payoffs, particularly in an environment with adversarial nodes. The incorporation of game theory into blockchain networks requires a thorough assessment of node strategies, the extent of information accessible, and the ensuing payoffs. It is noteworthy that peers' strategies and actions often modulate these payoffs. The game-theoretic approach provides a mathematical framework for modeling and analyzing strategic interactions among rational entities.

In the context of the proposed framework, nodes aim to optimize their rewards by adjusting the values within their trust matrices. However, these adjustments are not arbitrary. They are guided by rules and strategies that consider the actions of the other nodes. The application of game theory ensures that no single method dominates, preventing any node from accruing excessive rewards unfairly. Furthermore, nodes that consistently exhibit cooperative behavior or uphold the integrity of the network are more likely to be rewarded, thereby ensuring fairness in the reward distribution.

A theoretical game model is constructed, wherein the executors within a blockchain network are conceptualized as players. This model inherently aligns with the characteristics of cooperative games. Such an alignment is predicated on the rationale that nodes that function as collaborative agents are incentivized to maximize their reward outcomes collectively. Furthermore, this game is classified as a non-zero-sum game. This classification stems from the principle that rewards accrued by any individual node do not preclude the distribution of the gains among the other participating nodes. The non-zero-sum nature of the game intrinsically fosters a cooperative dynamic among the nodes as they endeavor to optimize their collective earnings while facilitating transaction execution and broadcast.

Moreover, the game is delineated as perfect information. This delineation is based on the premise that each node possesses comprehensive knowledge of the actions under-

taken by its counterparts. Such transparency is achieved via a truth matrix serving as a repository of all player actions, enabling informed decision-making processes. Given that this decision-making process and strategic evolution occur over multiple rounds, the game can be conceptualized as infinite. A detailed exposition and formalization of the game's characteristics, including its strategic dimensions and implications for blockchain network operations, will be elaborated upon in subsequent chapters of this paper.

2.5. Nash Equilibrium

The Nash Equilibrium, a foundational concept in game theory introduced by John Nash, represents a state in a strategic interaction where no player can unilaterally change their strategy to achieve a better outcome, given the other players' strategies. In more formal terms, a strategy set constitutes a Nash Equilibrium [10] if no player can obtain a higher payoff by deviating from their strategy, whereas other players keep their strategies unchanged. In blockchain networks, Nash Equilibrium can provide insightful perspectives regarding the stability of the strategy adopted by nodes. For instance, in a scenario where nodes decide whether to validate and broadcast transactions or act maliciously or honestly, a Nash Equilibrium would occur if no single node could unilaterally change its strategy (e.g., from honest to malicious or vice versa) to achieve a higher payoff, given the strategy adopted by the other nodes.

Thus, analyzing and ensuring scenarios in which honest behavior and cooperation among nodes form a Nash Equilibrium becomes pivotal, ensuring the robustness and security of the blockchain network among rational self-interested nodes. The proposed framework's primary objective is to engineer systemic convergence where all participating nodes within the blockchain network attain a Nash Equilibrium state. This desired equilibrium state underscores a strategic orientation wherein the actions and behaviors of the nodes collectively gravitate towards optimizing the gains for the entire network. Underpinning this strategic alignment is the principle that engaging in cooperative behavior is each node's most advantageous course of action. This cooperative dynamic was facilitated and reinforced by a meticulously designed system of rewards and penalties.

The game's rules are structured to guide the nodes towards this equilibrium. The system is designed to incentivize collaborative efforts and discourage actions detrimental to the network's collective interests. An in-depth exploration and analysis of the mechanisms underlying this reward/punishment system, along with a comprehensive discussion of how these elements synergistically contribute to achieving Nash Equilibrium within the blockchain network, will be presented in subsequent chapters of this paper.

3. Related Works

The establishment and implementation of robust incentive mechanisms is paramount in the intricate landscape of network systems, particularly those that are decentralized, such as blockchain networks. Several research endeavors have navigated various facets of incentive mechanisms, each presenting unique approaches and methodologies that could be insightful for developing and enhancing incentive models in blockchain networks.

3.1. Current Advances in Incentivizing Network Nodes

Li and Shen (2011) explored the application of game theory for analyzing cooperative incentive strategies in network systems [11]. They provide strategies whereby nodes determine their peers' trustworthiness, which can be adapted to blockchain networks to ensure that nodes can validate the reliability of peer transactions and blocks, thereby maintaining trust and reliability within the network.

Following this, Mahmoud and Shen (2012) introduced FESCIM [12], an incentive mechanism for multi-hop cellular networks that stimulated nodes to cooperate in packet forwarding by providing them with suitable incentives and ensuring fairness by preventing selfish nodes from depleting the network. When applied in a different network context, this approach could offer valuable insights into developing incentive mechanisms that ensure

fair resource utilization in blockchain networks, particularly in scenarios where nodes may be required to forward transactions or data to peers.

Subsequently, Dias et al. (2014) evaluated a cooperative reputation system for vehicular delay-tolerant networks and showed how reputation mechanisms can punish malicious nodes and reward well-behaved nodes [13]. This approach can be adapted to blockchain networks to ensure that nodes propagating valid transactions are rewarded. In contrast, those propagating invalid transactions are penalized, thereby maintaining a balance and ensuring that nodes are incentivized to act in the network's best interests.

Later, Yang et al. (2017) explored a social incentive mechanism to promote cooperation in mobile crowdsensing by leveraging social ties among participants [14]. This mechanism could be particularly innovative in blockchain environments where network participants have established social or economic ties, utilizing these pre-existing relationships to enhance network cooperation and integrity.

Finally, building on trust and reliability, He et al. (2018) proposed a blockchain-based truthful incentive mechanism for distributed P2P applications, demonstrating its effectiveness in stimulating nodes to share their unused resources using game analysis and evaluation [15]. This approach could be insightful in incentivizing nodes in blockchain networks to actively participate in transaction validation and block propagation, thereby ensuring that the network remains active and secure.

In conclusion, while varying in their application and context, these studies present a rich tapestry of approaches toward incentivizing node cooperation and participation in various network systems.

3.2. Incentive Models in Decentralized Blockchain Networks

In the context of a decentralized blockchain network such as Ethereum, the processing and validation of transactions rely on the active participation of nodes [16]. Nodes, whether people or organizations, donate computing resources, time, and energy to carry out transaction-processing responsibilities. Nevertheless, the guaranteeing of active and sincere engagement from the nodes is a significant obstacle. With suitable incentives, nodes may be willing to participate actively in transaction-processing endeavors. The lack of incentives may give rise to several challenges, such as decreased efficiency in transaction processing, compromised network security, and the dissemination of fraudulent transactions by malicious entities.

The provision of incentives to nodes in a blockchain network has distinct issues compared with conventional centralized systems [17]. In centralized systems, it is possible for a central authority to implement incentives and penalties effectively. In a decentralized blockchain setting, the lack of a central governing body requires the establishment of incentive frameworks that harmonize individual nodes' motivations with the network's overall well-being. This alignment is paramount to guarantee the nodes' intended functionality and preserve the network's integrity. Furthermore, it is essential for the incentive model to effectively tackle the difficulties associated with fairness, scalability, and security, as these factors significantly affect the efficient operation of blockchain networks. The preservation of a robust and secure blockchain network requires active and sincere engagement with the nodes.

This section explores the need for incentivizing the nodes to maintain their ongoing participation in transaction processing and network maintenance. By offering suitable incentives, nodes can be motivated to allocate their computing resources, time, and energy to maintain the blockchain network. The incentivization process is of utmost importance in pursuing consensus, enhancing transaction processing speed, bolstering network security, and promoting the general resilience of the blockchain ecosystem. This section examines the incentive mechanisms currently used in blockchain systems, with a specific emphasis on two critical methodologies: Proof of Work (PoW) [4] and Proof of Stake (PoS) [16]. However, this analysis acknowledges that such incentivization predominantly targets miner nodes rather than executor nodes, which are the primary focus of our re-

search study. Consequently, our exploration delves into the implications and outcomes of these incentive mechanisms specifically as they pertain to the executor nodes within the blockchain network.

3.2.1. Proof of Work (PoW)

The proof of work (PoW) is a cryptographic puzzle-solving system in which miners compete to solve complex mathematical problems. The first miner to solve the cryptographic challenge may recommend a second block of transactions to the blockchain. Despite its high resource requirements, this technique validates transactions, improves network security, and encourages participation [18]. Miners are motivated to solve cryptographic problems using the possibility of a block reward. A payout generally includes newly created cryptocurrency tokens and transaction fees from the block transactions. Currency generation and transaction fee collection create a dynamic environment where miners are financially incentivized to validate transactions and secure the network. The validation process involves miners from many backgrounds and regions, preventing one entity from dominating the network and creating a more democratic and healthy ecosystem. Figure 3 illustrates a high-level overview of the workflow involved in the Proof of Work system, providing a visual representation of the process described above.

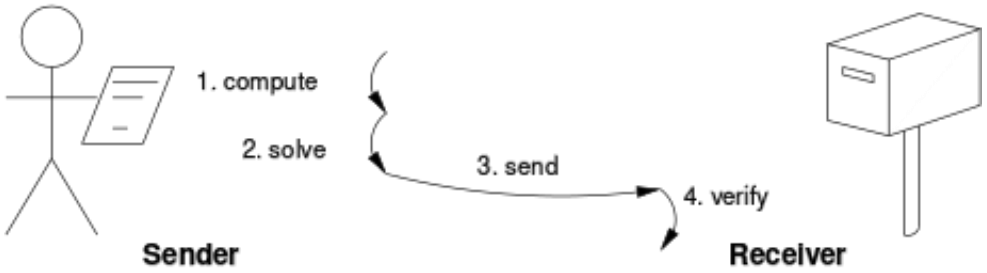

Figure 3. High-level overview of workflow of PoW.

3.2.2. Proof of Stake (PoS)

The proof of stake (PoS) operates on a different premise than the proof of work (PoW). Proof of stake (PoS) uses computational puzzle-solving to choose block validators based on the number of tokens they stake as collateral [19]. Token ownership and other factors affect validator selection, which involves proposing and confirming new blocks. Node incentivization is essential for the proof of stake (PoS). Block validators are motivated by transactions and block rewards. This incentive structure aligns with network security because validators invest financially in blockchain integrity.

3.2.3. Alternative Approaches in Reputation-Based Blockchain Consensus

Alzahrani and Bulusu [20], in 2018, introduced a groundbreaking blockchain consensus protocol that merged game theory with randomness to achieve true decentralization, steering clear of traditional Proof of Work (PoW) methods. Their innovative protocol, designed to select varying validator sets randomly for each block, employs game-theoretical models to encourage honest behavior among participants. This novel approach significantly bolsters security, particularly against attacks like DDoS and eclipse attacks, marking a substantial leap in blockchain technology, particularly in enhancing security and efficiency in a more dynamic and secure blockchain environment.

In 2019, Yun, Goh, and Chung [21] proposed the Trust-Based Shard Distribution (TBSD) scheme, focusing on elevating the security of blockchain systems within sharded networks. TBSD effectively allocates nodes across shards based on trust assessments, thereby curtailing the likelihood of malicious node concentrations. The scheme aims to achieve fairness and reduce discrepancies in shard distribution by incorporating a genetic algorithm. The effectiveness of TBSD in preserving blockchain network integrity, particularly against potential attacks, underscores its emphasis on equitable and secure shard distribution.

Then, in 2020, Huang et al. introduced RepChain [22], a sharding-based blockchain system that harnessed reputation scores to overcome limitations in existing sharding frameworks. RepChain's unique double-chain architecture, which includes transaction and reputation chains, aims to boost throughput and security and incentivize node cooperation. The system employs Raft-based synchronous consensus for transactions and Byzantine fault tolerance for reputation management, ensuring effective processing and heightened security. Theoretical analyses and evaluations on Amazon Web Service platforms demonstrate RepChain's capacity to enhance throughput and security in sharding-based blockchain systems.

In 2022, Qiu et al. presented a dynamic reputation-based consensus mechanism for blockchain [23], addressing centralization issues inherent in Proof-of-Authority (PoA) systems. Their mechanism introduces a reputation evaluation algorithm for selecting high-reputation nodes as validators, thereby deterring malicious behavior. Monitoring nodes are utilized to oversee validators and safeguard the network. The proposed mechanism notably improves fault tolerance, expedites consensus time, and bolsters system security. However, the paper calls for further research in node evaluation optimization due to the potential impact of malicious credit evaluations on node credibility.

Moving forward to 2023, several significant contributions were made in the field. Xiang Li et al. proposed the GTI mechanism [24], aligned with the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm, that incentivized honest behavior among validators via auditing and penalized lazy nodes via deposit loss, addressing challenges in non-cryptocurrency blockchains.

Furthermore, Xiao Liu et al. introduced a method leveraging evolutionary game theory [25] to mitigate block withholding attacks in blockchain systems. Their dynamic game model adapts to the system's degree of supervision and punishment, offering insights into optimal strategies for mining pools under various attack scenarios.

Also, in 2023, Jauzak Hussaini et al. introduced the Proof of Intelligent Reputation (PoIR) consensus mechanism [26]. PoIR combines BiLSTM with the Network Entity Reputation Database to generate reputation scores, selecting authoritative nodes for blockchain consensus. This mechanism shows enhanced resistance to centralization and efficiency in transaction times.

In conclusion, from 2018 to 2023, as summarized comprehensively in Table 2, these studies collectively present various approaches encompassing performance-based and reputation-based blockchain consensus algorithms. From game theory and randomness to advanced reputation-based methods, each contributes uniquely to advancing blockchain consensus mechanisms, reflecting a significant evolution in the field.

Table 2. Overview of recent research in performance-based and reputation-based blockchain consensus algorithms.

Consensus	Type and Selection	Players	Reward System	Dynamic Punishment Coefficient	MPC Compatible	51% Attack Resistant	Sybil Attack Resistant	BWH Resistant	Decentralized
Our Proposed Consensus (2024)	Reputation-based Deterministic assessment of node reputation	Executors Validators Miners	Dynamic rewards based on contribution	√	√	√	√	√	√
Game theory-based compatible incentive mechanism design for non-cryptocurrency blockchain systems (2023) [24]	Reputation-based Probabilistic assessment of node reputation	Validators	Equitably distributed across participants	-	-	√	-	-	√
An evolutionary game theory-based method to mitigate block withholding attack in blockchain system (2023) [25]	Incentive game theory Deterministic pool incentives	Miners	Equitably distributed across participants	-	-	√	-	√	-
PoIR: A Node selection mechanism in reputation-based blockchain consensus using bidirectional LSTM regression model (2023) [26]	Reputation-based Probabilistic assessment of node reputation	Validators	-	-	-	-	-	-	√
A dynamic reputation-based consensus mechanism for blockchain (2022) [23]	Reputation-based Primary election in PBFT	Validators	Favors personal over collective incentives	-	-	-	-	√	-
RepChain: A reputation-based, secure, fast, and high Incentive blockchain system via sharding (2020) [22]	Reputation-based via sharding Probabilistic assessment of node reputation	Validators	Static value	-	√	-	-	-	-
Trust-based shard distribution scheme for fault-tolerant shard blockchain networks (2019) [21]	Reputation-based Node trust assessment via genetic algorithm	Miners	-	-	-	√	-	-	-
Proof of stake (2019) [19]	Reputation-based with collaterals Cumulative stake amount	Validators	Accrues transaction fees	-	-	√	-	-	-
Towards true decentralization: A blockchain consensus protocol based on game theory and randomness (2018) [20]	Reputation-based with random selection Probabilistic assessment of node reputation	Validators	Static value	-	-	-	-	√	√
Proof of work (2008) [18]	Performance-based Cryptographic puzzle	Miners	Accrues transaction fees	-	-	-	-	-	√

4. Graph Modeling of Blockchain Nodes and Our Game Framework

Blockchain networks consist of several nodes engaging in interactions and reaching choices guided by self-interests. In the initial phase of developing our framework, it is essential to construct a mathematical model to facilitate a deeper understanding of the interactions among nodes within the blockchain network. We chose to represent the network using a graph-based model owing to graphs' versatile properties and structural intricacies. Specifically, we opted for an undirected graph to illustrate node communication dynamics effectively. Furthermore, the representation of the graph in a matrix form simplifies its depiction and enhances its integration into our algorithmic constructs. Subsequent stages involve incorporating this matrix into the strategic game, meticulously formalized in the forthcoming chapter. This integration enables nodes to make informed decisions in each round and adjust their trust matrices accordingly. The final step involves the implementation of an incentive mechanism designed to reward nodes proportionally based on their honest contributions and overall participation in the network. Figure 4 provides a high-level overview of this entire process, illustrating the key stages and how they interconnect within our framework.

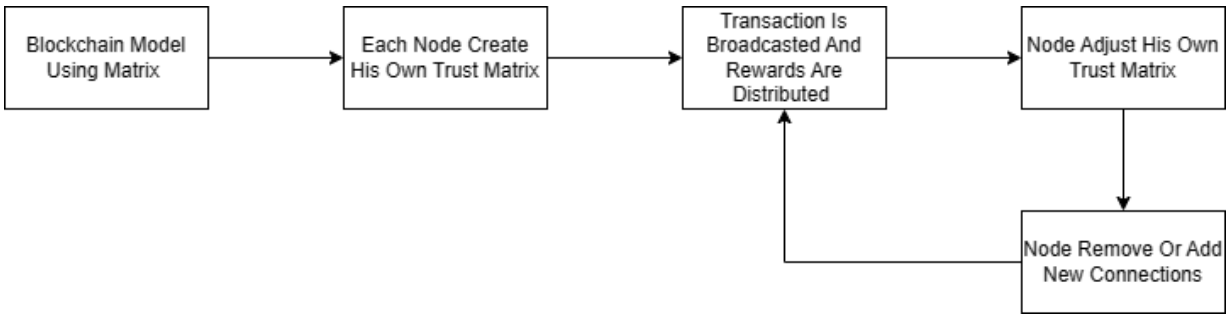

Figure 4. High-level overview of our process.

This chapter uses graph theory [27] to represent the node graph in a blockchain network while also integrating the consideration of node behavior probabilities. Each node is a vertex in the graph, with the edges representing communication between nodes. In addition, we used the concept of probabilities to denote the probability of a node being malicious or honest.

4.1. Modeling Nodes Graph with Probabilistic Behavior

In the context of a blockchain network, it is essential to consider the potential of nodes to exhibit malicious behavior. The network representation of each node remains the same, and each node is represented as a vertex. However, the edges in the graph indicate the chance of communication between the nodes. The probabilities mentioned may be interpreted as the probability of a transaction being communicated between nodes, considering the possibility of malicious activity. By integrating probabilistic behavior, we can effectively model the dynamics and inherent uncertainties that arise from the interactions between nodes in a network. Figure 5 illustrates an example of such a graph network in a blockchain, depicting a network with six nodes.

Using a matrix to represent each node's strategy is grounded in the need to systematically model and analyze complex probabilistic interactions in a blockchain network. The matrix format allows for a clear and organized representation of the intricate dynamics at play, where each element captures the probability of interaction between two nodes. This provides a structured overview of the network's inter-node communication patterns and sets the stage for applying game-theoretic techniques.

In game theory, matrices are often employed as payoff matrices that define the rewards or costs associated with the different strategic interactions between players. By representing blockchain node strategies and interactions in a matrix, we pave the way for applying

similar game-theory analyses. The goal is to predict and influence node behaviors in the network, particularly in the presence of potential adversarial actions. Thus, the matrix is foundational for studying and optimizing strategic interactions in a blockchain context.

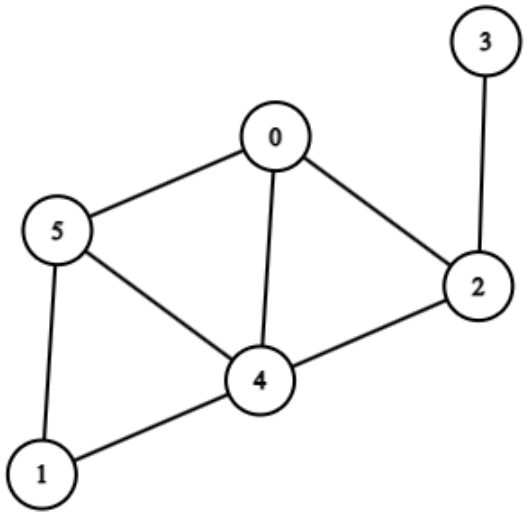

Figure 5. Graph network of a blockchain with 6 nodes.

In the above example, the graph can be represented using the following adjacency matrix.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

For a 6×6 matrix representing a graph, the diagonal elements of the matrix are consistently zero. This is because of the assumption that a node inside the network is unable to interact with itself. Moreover, it lacks any motivation to engage in fraudulent transactions with itself.

Probabilistic graph representation must be transformed into a matrix format to examine and manipulate the probabilistic characteristics of the nodes in the graph. Like the matrix structure, every node is associated with the matrix's columns and rows. Nevertheless, in contrast to using binary values (0 or 1) to denote the presence or absence of communication, the matrix includes probabilities ranging from 0 to 1, which signifies the likelihood or possibility of communication. The matrix offers succinct and organized depictions of the probabilistic communication patterns in the blockchain network. This modeling methodology allows the assessment of a network's resilience and robustness in the face of malicious actions. Furthermore, the integration of probabilities enables the examination of tactics aimed at reducing the consequences of malicious acts and encouraging truthful engagement, thus cultivating a blockchain network that is more robust and dependable. The last graph can be represented using updated values.

$$\begin{pmatrix} 0 & 0 & 0.4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.3 & 1 \\ 1 & 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0.5 & 1 & 0 & 0 & 1 \\ 0.7 & 0.4 & 0 & 0 & 0.1 & 0 \end{pmatrix}. \tag{2}$$

Each node has its matrix, called the trust matrix, which is determined by the strategy it aims to apply. The starting values of the matrix are established accordingly. The ultimate objective is implementing a sequence of operations on the matrix to maximize the individual's node benefits. One possible solution is introducing a node that consistently employs a cooperative approach [28], resulting in a specific matrix configuration.

$$\begin{pmatrix} 0 & 1 & . & . & 1 & 1 \\ 1 & . & . & . & . & 1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & . & . & . & . & 1 \\ 1 & 1 & . & . & 1 & 0 \end{pmatrix} \tag{3}$$

An additional node may be included to initiate a cooperative approach and modify its values in response to the round outcome. The node can use either a pure or a mixed strategy [29]. This analysis examines each strategy and assesses its applicability in a specific situation.

4.2. Game Framework and Rules

In the initial phase of our research, we focused on conceptualizing interactions within a blockchain network using graph theory and matrix representations. This section aims to construct a strategic game, facilitating nodes to adjust their trust matrix and make strategic decisions over successive rounds. These decisions include creating or removing connections with other nodes based on evolving strategies. We begin by defining the critical elements of the framework as follows.

- Set of Players: This is represented by a finite set of nodes within the graph, denoted as $N = \{1, 2, \ldots, n\}$, where each node is a player in the game. These players can exhibit malicious or honest behaviors depending on their chosen strategies. To capture the dynamic nature of the network and align with real-world scenarios, a node is permitted to switch strategies across rounds, thus enabling both cooperative and non-cooperative behaviors.
- Actions/Strategies: Nodes have two fundamental types of actions. The first relates to handling transactions or messages, where a node decides to either forward it to the next node or refrain from doing so, essentially choosing to cooperate. This can be represented by a binary sequence, such as $\{00011 \ldots 00\}$. For simplicity, we initially assume that a node applies the same action uniformly across all nodes, although the model retains the flexibility to account for varying actions in future iterations. The second action concerns modifying relationships with other nodes, influenced by their strategy and the trust matrix. This could entail removing connections with nodes that fall below a specific reputation score, as determined by the trust matrix.
- Payoffs: Each node's actions result in a reward or a penalty. Actively participating in transaction broadcasting yields rewards proportionate to the node's contribution to the network's connectivity and transaction throughput. In contrast, non-cooperative behavior reduces the node's reputation, potentially leading to isolation within the network. For this process to be fair, a node will be rewarded based on his weight and contribution to the network, meaning that a node that broadcasted to three nodes will not be rewarded the same as one that broadcasted to ten nodes; having this logic will incentivize nodes to prefer to be hubs in the network. The Reward System chapter provides a detailed exposition of this mechanism.
- Order of Play: The first player in each round is determined by the initiator of the transaction, essentially chosen using a pseudo-random process based on transaction initiation. The selected node then propagates the transaction to other nodes, akin to a token being passed in a ring topology. The game possesses several distinct characteristics:

- Cooperative Nature: The framework views nodes as components of potential coalitions, with an ideal scenario being a single unified coalition. Represented as (N, v), the game promotes common interests by focusing on a network's well-being and security. The power set 2^n encapsulates all possible coalitions, for example, for a graph of three nodes.

$$2^3 = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\} \tag{4}$$

The Characteristic function $v(C)$ is defined below.

$$\left(\forall \mathrm{C} \in 2^{\mathrm{N}}/\emptyset\right) \mathrm{v}(\mathrm{C}) = \mathrm{R} \tag{5}$$

where R is the fixed total reward for each stage, and C is an element of 2^N. At first sight, it appears that the most optimal strategy for a node is to be alone in its coalition. However, because the game is repetitive, acting with this strategy may be better because it does not encourage nodes to participate in the group's gain. The design incentivizes stable and fair coalitions, where no subset of players is incentivized to form a smaller coalition, and each player's payoff is proportional to their contribution.

- Repetitive Game Structure: Denoted as G^T, the game is repetitive, with players selecting strategies at each stage of the game iteration. The cumulative payoff is the sum of the rewards obtained across all stages, and strategies are adapted based on the history of the play, allowing dynamic responses to the outcomes of previous rounds.
- Non-Zero-Sum Nature: The game is designed as a non-zero-sum game. If $u_i(s_i)$ denotes the payoff for choosing a strategy, then no combination of strategies results in a total payoff sum of zero.

$$\sum_{i=1}^{n} u_i(s_1, s_2, \ldots, s_n) \neq 0 \tag{6}$$

Thus, our model offers a nuanced approach to understanding and guiding node behavior in blockchain networks by considering individual strategies and collective network dynamics. Figure 6 provides an example of a graph network in a blockchain with three nodes, illustrating a simplified version of such a network and the possible connections between nodes. Note that this framework has the following constraints.

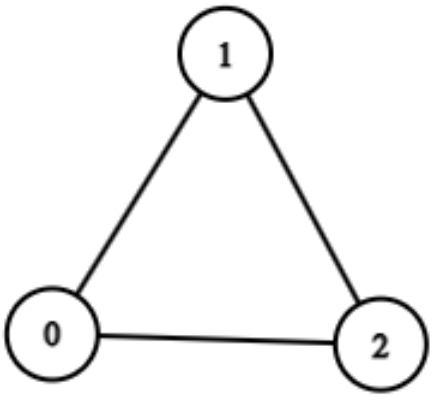

Figure 6. Graph network of a blockchain with 3 nodes.

1. We should avoid selfish mining [30] and withholding attacks [31]. Selfish mining and withholding attacks are strategies miners use in blockchain networks to gain an unfair advantage and potentially disrupt network functioning. In selfish mining, a miner or group of miners secretly mines blocks but withholds broadcasting them to the network. They continue to mine the next block, creating a longer private chain than the public one. When another miner solves a block and threatens to catch up, the

selfish miner releases their previously hidden blocks, making their chain the longest and most accepted one, rendering the honest miners' efforts useless. This wastes the computational resources of other miners and can lead to increased centralization and reduced network security [32]. Both attacks undermine the fundamental principles of fair and decentralized mining inherent to blockchain technology.

2. The network should be dynamic, meaning that the nodes should constantly create or remove new connections; this mechanism is explained in chapter 7. Actions Based on The Trust Matrix.
3. The network should contain N nodes that verify the following constraint $N \geq 3f + 1$ and f is the number of faulty nodes, an inherited constraint owing to the use of the PBFT algorithm [33].

5. Our Innovative Framework for Node Incentivization and Trust Optimization in Blockchain Networks

Each node will have its trust matrix; we can represent the matrix by the following denotation S_{nodeID}^{genNum} where nodeID is the id of the node in the graph and the genNum is the functional power of the application f that the node will apply each generation/round; therefore, if we are noting S_1^3, we are referring to the matrix of the node 1 in round 3, and the way it is generated is the following.

$$S_1^3 = f\left(S_1^2\right) = fof(S_1^1) = fofof\left(S_1^0\right) = f^3\left(S_1^0\right) \tag{7}$$

In general, we have the following formula.

$$S_{nodeID}^{genNum} = f^{genNum}\left(S_{nodeID}^0\right) \tag{8}$$

We will also note $a_{i,j}$ elements of the S_{nodeID}^{genNum} matrix.

The strategy starts by generating a matrix with the number of nodes in the network, which can be reduced to a non-zero-sum game using the following steps:

1. In the first round, the node generates a matrix in order of the number of nodes in the network.
2. The node broadcasts its matrix to other nodes using a Practical Byzantine Fault Tolerance (PBFT) algorithm (PBFT (Practical Byzantine Fault Tolerance): An algorithm used in computer systems to reach agreement (consensus) in a network of unreliable processors. In the context of blockchain, PBFT is often utilized to achieve consensus in a decentralized environment, ensuring that nodes agree on the validity of transactions even in the presence of malicious nodes).
3. In the second round, the node generates another matrix with new values based on the old matrix and the rewards generated in the last round to maximize the rewards.
4. Node can also remove other nodes and alter the values based on the matrix of the other nodes.

In that case, we will have the following trust matrix.

$$S_0^0 = \begin{pmatrix} 0 & a_{01}^0 & a_{02}^0 \\ a_{10}^0 & 0 & a_{12}^0 \\ a_{20}^0 & a_{21}^0 & 0 \end{pmatrix} S_1^0 = \begin{pmatrix} 0 & b_{01}^0 & b_{02}^0 \\ b_{10}^0 & 0 & b_{12}^0 \\ b_{20}^0 & b_{21}^0 & 0 \end{pmatrix} S_2^0 = \begin{pmatrix} 0 & c_{01}^0 & c_{02}^0 \\ c_{10}^0 & 0 & c_{12}^0 \\ c_{20}^0 & c_{21}^0 & 0 \end{pmatrix} \tag{9}$$

We can imagine that the three nodes will start with these matrices, having the first node trusting everyone, and from the start will place every element at 1, the second node will have a probability of 0.5, and the third node will not trust anyone.

$$S_0^0 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} S_1^0 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} S_2^0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

For transactions received from node 0 and because he is trusting everyone, he will broadcast the transaction to the entire network; if all nodes are not malicious, each node will broadcast the same transaction and store it, the three nodes will sign the transaction, and the rewards will be divided among the three nodes. The truth matrix is expressed as follows:

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{11}$$

Each node applies the following transformation to increase the reputation score of another node.

$$\left(\forall a_{i,j} \in S\right) f\left(a_{i,j}\right) = \frac{a_{i,j} + 1}{2} \tag{12}$$

Therefore, the matrix of the next generation will be

$$S_0^1 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} S_1^1 = \begin{pmatrix} 0 & 0.75 & 0.75 \\ 0.75 & 0 & 0.75 \\ 0.75 & 0.75 & 0 \end{pmatrix} S_2^1 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} \tag{13}$$

The same logic applies to all future generations.

$$S_0^2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} S_1^2 = \begin{pmatrix} 0 & 0.875 & 0.875 \\ 0.875 & 0 & 0.875 \\ 0.875 & 0.875 & 0 \end{pmatrix} S_2^2 = \begin{pmatrix} 0 & 0.75 & 0.75 \\ 0.75 & 0 & 0.75 \\ 0.75 & 0.75 & 0 \end{pmatrix} \tag{14}$$

And we can generalize for a n generation.

$$\begin{gathered} S_0^n = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} S_1^n = \begin{pmatrix} 0 & \frac{0.5+2^n-1}{2^n} & \frac{0.5+2^n-1}{2^n} \\ \frac{0.5+2^n-1}{2^n} & 0 & \frac{0.5+2^n-1}{2^n} \\ \frac{0.5+2^n-1}{2^n} & \frac{0.5+2^n-1}{2^n} & 0 \end{pmatrix} \\ S_2^n = \begin{pmatrix} 0 & \frac{0+2^n-1}{2^n} & \frac{0+2^n-1}{2^n} \\ \frac{0+2^n-1}{2^n} & 0 & \frac{0+2^n-1}{2^n} \\ \frac{0+2^n-1}{2^n} & \frac{0+2^n-1}{2^n} & 0 \end{pmatrix} \end{gathered} \tag{15}$$

$$\begin{gathered} S_0^n = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} S_1^n = \begin{pmatrix} 0 & \frac{-0.5}{2^n} + 1 & \frac{-0.5}{2^n} + 1 \\ \frac{-0.5}{2^n} + 1 & 0 & \frac{-0.5}{2^n} + 1 \\ \frac{-0.5}{2^n} + 1 & \frac{-0.5}{2^n} + 1 & 0 \end{pmatrix} \\ S_2^n = \begin{pmatrix} 0 & 1 - \frac{1}{2^n} & 1 - \frac{1}{2^n} \\ 1 - \frac{1}{2^n} & 0 & 1 - \frac{1}{2^n} \\ 1 - \frac{1}{2^n} & 1 - \frac{1}{2^n} & 0 \end{pmatrix} \end{gathered} \tag{16}$$

$$\lim_{n \to +\infty} S_0^n = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \lim_{n \to +\infty} S_1^n = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \lim_{n \to +\infty} S_2^n = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \tag{17}$$

$$\lim_{n \to +\infty} S_0^n = \lim_{n \to +\infty} S_1^n = \lim_{n \to +\infty} S_2^n \tag{18}$$

If two nodes are malicious (Nodes 1 and 2, for example), the truth matrix will be the following.

$$\begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \tag{19}$$

Malicious nodes were detected using the PBFT algorithm. Each node applies the following transformation if it wants to decrease the score reputation of another node:

$$\left(\forall a_{i,j} \in S/i \neq j\right) f\left(a_{i,j}\right) = \frac{a_{i,j}}{2} \tag{20}$$

Therefore, the matrix of the next generation will be:

$$S_0^1 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} S_1^1 = \begin{pmatrix} 0 & 0.25 & 0.25 \\ 025 & 0 & 0.25 \\ 0.25 & 0.25 & 0 \end{pmatrix} S_2^1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{21}$$

The same logic applies to all future generations:

$$S_0^2 = \begin{pmatrix} 0 & 0.25 & 0.25 \\ 0.25 & 0 & 0.25 \\ 0.25 & 0.25 & 0 \end{pmatrix} S_1^2 = \begin{pmatrix} 0 & 0.125 & 0.125 \\ 0.125 & 0 & 0.125 \\ 0.125 & 0.125 & 0 \end{pmatrix} S_2^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{22}$$

And we can generalize for an N generation.

$$S_0^n = \begin{pmatrix} 0 & \frac{1}{2^n} & \frac{1}{2^n} \\ \frac{1}{2^n} & 0 & \frac{1}{2^n} \\ \frac{1}{2^n} & \frac{1}{2^n} & 0 \end{pmatrix} S_1^n = \begin{pmatrix} 0 & \frac{0.5}{2^n} & \frac{0.5}{2^n} \\ \frac{0.5}{2^n} & 0 & \frac{0.5}{2^n} \\ \frac{0.5}{2^n} & \frac{0.5}{2^n} & 0 \end{pmatrix} S_2^n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{23}$$

$$\lim_{n\to+\infty} S_0^n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \lim_{n\to+\infty} S_1^n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \lim_{n\to+\infty} S_2^n = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{24}$$

$$\lim_{n\to+\infty} S_0^n = \lim_{n\to+\infty} S_1^n = \lim_{n\to+\infty} S_2^n \tag{25}$$

If two nodes are malicious (Node 2, for example): The truth matrix will be the following.

$$\begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \tag{26}$$

Each node applies the following transformation, a combination of Equations (12) and (20), where $s_{i,j}$ represents the coefficients of the truth matrix.

$$f\left(a_{i,j}\right) = \begin{cases} \frac{a_{i,j}}{2} & s_{i,j} = -1 \\ \frac{a_{i,j}+1}{2} & s_{i,j} = 1 \end{cases} = \frac{a_{i,j} + \frac{s_{i,j}+1}{2}}{2} = \frac{2a_{i,j} + s_{i,j} + 1}{4} \tag{27}$$

Therefore, the matrix of the next generation will be

$$S_0^1 = \begin{pmatrix} 0 & 1 & 0.5 \\ 1 & 0 & 1 \\ 0.5 & 1 & 0 \end{pmatrix} S_1^1 = \begin{pmatrix} 0 & 0.75 & 0.25 \\ 075 & 0 & 0.75 \\ 0.25 & 0.75 & 0 \end{pmatrix} S_2^1 = \begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{pmatrix} \tag{28}$$

The same logic applies to all future generations.

$$S_0^2 = \begin{pmatrix} 0 & 1 & 0.25 \\ 1 & 0 & 1 \\ 0.25 & 1 & 0 \end{pmatrix} S_1^2 = \begin{pmatrix} 0 & 0.875 & 0.125 \\ 0.875 & 0 & 0.875 \\ 0.125 & 0.875 & 0 \end{pmatrix} S_2^2 = \begin{pmatrix} 0 & 0.75 & 0 \\ 0.75 & 0 & 0.75 \\ 0 & 0.75 & 0 \end{pmatrix} \quad (29)$$

And we can generalize for a n generation.

$$S_0^n = \begin{pmatrix} 0 & 1 & \frac{1}{2^n} \\ 1 & 0 & 1 \\ \frac{1}{2^n} & 1 & 0 \end{pmatrix} S_1^n = \begin{pmatrix} 0 & \frac{0.5+2^n-1}{2^n} & \frac{0.5}{2^n} \\ \frac{0.5+2^n-1}{2^n} & 0 & \frac{0.5+2^n-1}{2^n} \\ \frac{0.5}{2^n} & \frac{0.5+2^n-1}{2^n} & 0 \end{pmatrix}$$
$$S_2^n = \begin{pmatrix} 0 & \frac{0.5+2^n-1}{2^n} & 0 \\ \frac{0.5+2^n-1}{2^n} & 0 & \frac{0.5+2^n-1}{2^n} \\ 0 & \frac{0.5+2^n-1}{2^n} & 0 \end{pmatrix} \quad (30)$$

$$\lim_{n\to+\infty} S_0^n = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \lim_{n\to+\infty} S_1^n = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \lim_{n\to+\infty} S_2^n = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (31)$$

$$\lim_{n\to+\infty} S_0^n = \lim_{n\to+\infty} S_1^n = \lim_{n\to+\infty} S_2^n \quad (32)$$

In this example, the trust matrix is a simpler version; if you are among the nodes that contribute to the broadcast operation, you will get a trust bonus of 1, and if you do not cooperate, you will be punished by -1. However, the matrix can be adjusted and has two different coefficients: the coefficient of reward and the coefficient of punishment. Therefore, we can generalize the application as follows, where c_p is the coefficient of punishment, c_r is the coefficient of reward, and $s_{i,j}$ are the elements of the truth matrix:

$$\left(\forall a_{i,j} \in S / i \neq j\right)\left(c_p \in [1;+\infty[\right)(c_r \in]0;1]\right) f\left(a_{i,j}\right) = \begin{cases} \frac{a_{i,j}}{2^{c_p}} & s_{i,j} = -1 \\ \frac{a_{i,j}+c_r}{2} & s_{i,j} = 1 \end{cases} \quad (33)$$

Each node can have its trust matrix based on the risk tolerance. If we can extend the formula for N-generation, we have:

$$f^n\left(a_{i,j}\right) = \begin{cases} \frac{a_{i,j}}{2^{n*c_p}} & s_{i,j} = -1 \\ \frac{a_{i,j}+(2^n-1)c_r}{2^n} & s_{i,j} = 1 \end{cases} \quad (34)$$

Therefore:

$$\lim_{n\to+\infty} f^n\left(a_{i,j}\right) = \begin{cases} 0 & s_{i,j} = -1 \\ c_r & s_{i,j} = 1 \end{cases} \quad (35)$$

Proposition 1. *We should have, if all nodes are not malicious and use the same reward coefficient:*

$$\forall\, i \in [0, |V| - 1] \lim_{n\to+\infty} S_i^n = \lim_{n\to+\infty} S_{i+1}^n \quad (36)$$

Proof 1. Let us have two nodes, i and $i + 1$; their matrices are S_i^n and S_{i+1}^n for n round.

If $a_{i,j}$, $b_{i,j}$ are the coefficients of their respective matrices, we will have the following.

$$S_i^n = \begin{pmatrix} 0 & \cdots & \frac{a_{1,d}+(2^n-1)c_r}{2^n} \\ \vdots & \ddots & \vdots \\ \frac{a_{d,1}+(2^n-1)c_r}{2^n} & \cdots & 0 \end{pmatrix} \quad (37)$$

$$S_{i+1}^{n} = \begin{pmatrix} 0 & \cdots & \frac{b_{1,d}+(2^n-1)c_r}{2^n} \\ \vdots & \ddots & \vdots \\ \frac{b_{d,1}+(2^n-1)c_r}{2^n} & \cdots & 0 \end{pmatrix} \tag{38}$$

$$\lim_{n\to+\infty} S_i^n = \lim_{n\to+\infty} \begin{pmatrix} 0 & \cdots & \frac{a_{1,d}+(2^n-1)c_r}{2^n} \\ \vdots & \ddots & \vdots \\ \frac{a_{d,1}+(2^n-1)c_r}{2^n} & \cdots & 0 \end{pmatrix} = \begin{pmatrix} 0 & \cdots & c_r \\ \vdots & \ddots & \vdots \\ c_r & \cdots & 0 \end{pmatrix} \tag{39}$$

$$\lim_{n\to+\infty} S_{i+1}^n = \lim_{n\to+\infty} \begin{pmatrix} 0 & \cdots & \frac{b_{1,d}+(2^n-1)c_r}{2^n} \\ \vdots & \ddots & \vdots \\ \frac{b_{d,1}+(2^n-1)c_r}{2^n} & \cdots & 0 \end{pmatrix} = \begin{pmatrix} 0 & \cdots & c_r \\ \vdots & \ddots & \vdots \\ c_r & \cdots & 0 \end{pmatrix} \tag{40}$$

$$\text{Therefore, } \lim_{n\to+\infty} S_i^n = \lim_{n\to+\infty} S_{i+1}^n \tag{41}$$

□

6. Reward System

The goal is to create a reward system that provides incentive rewards to the nodes and encourages nodes to broadcast to more nodes. The simple version of sharing rewards is to divide the rewards among the contributors' nodes equally; in that case, we will have $R_i = \frac{R_t}{N}$ where R_i is the reward for the Node(i), R_t is the total reward, and N is the number of contributor nodes. However, if we use this reward system in the following network, the nodes will fail to share with other nodes and the node will forward traffic to only one node. In this case, node 0 can broadcast only to node 1 and receives the same rewards as broadcasting to nodes 0 and 1; therefore, the optimal strategy is to broadcast to only one node, which is not the case we aim for. Figure 7 illustrates a graph network of a blockchain with eight nodes, showing a potential network configuration where such reward dynamics might occur. Each topology has its specificity, and different reward mechanisms should be applied for each topology. We can deduce that the reward graph is a subgraph from the initial network where nodes are the contributor nodes; it is a directed graph starting from the first node that broadcasts the transaction, and the graph represents the transaction flow in the graph. Figure 8 presents an example of a reward tree represented as a subgraph of the network, illustrating how rewards can be distributed among the contributor nodes.

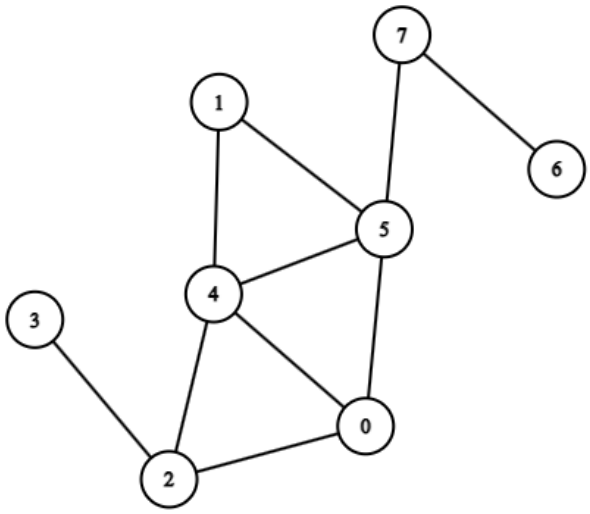

Figure 7. Graph network of a blockchain with 8 nodes.

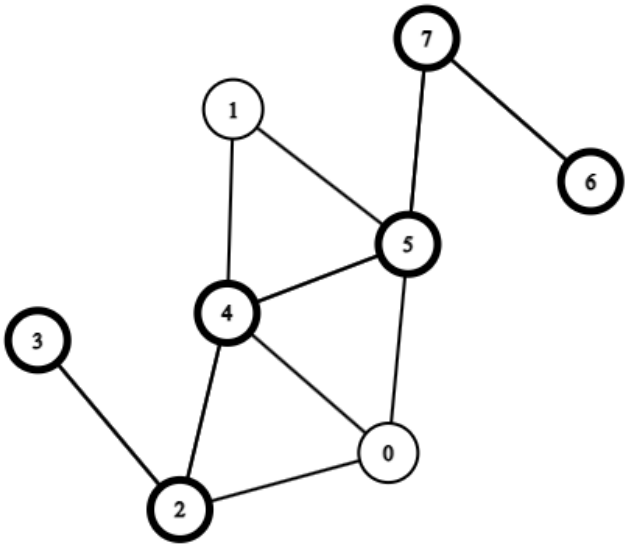

Figure 8. Example of a reward tree represented as a subgraph of the network.

In this example, the reward graph is the subgraph {3,2,4,5,7,6}. The goal is to present a fair reward system for all the nodes. That is, the number of nodes is not the only factor in the distribution, but also how the reward graph is structured; the more connections a node has, the more rewards it will have. Furthermore, if a node has a lower rank in the graph, the reward will be higher because it is among the first broadcasters of the transaction. Figure 9 depicts an example of a reward tree graph composed of six nodes, demonstrating how different nodes, based on their position and connections within the network, receive varying rewards.

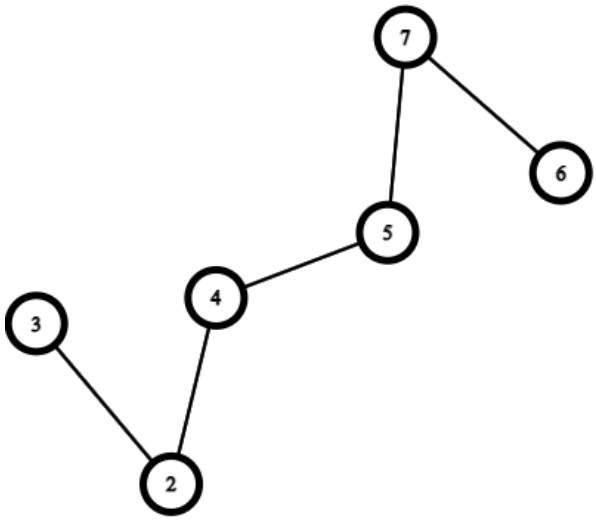

Figure 9. Example of a reward tree graph composed of 6 nodes.

The following graphs have the same degree; however, their structures differ. The logic we will apply is that each node will have a weight based on its contribution. For example, in all graphs in Figure 10, if node 0 does not broadcast the transaction, no node receives it. Therefore, node 0 makes a significant contribution to the network. To apply this logic, we counted the weight as the number of children and sub-children of that node using the depth-first search [34] algorithm (DFS) (Depth First Search (DFS): A fundamental graph traversal algorithm that explores as far as possible along each branch before backtracking. It allows for comprehensive exploration of all vertices and edges, making it effective in determining child and sub-child nodes for a given node in the context of the described

weighting logic). When we obtain that value, we normalize it by dividing it by the sum of all other weights. The following is a simple example of a graph with eight nodes.

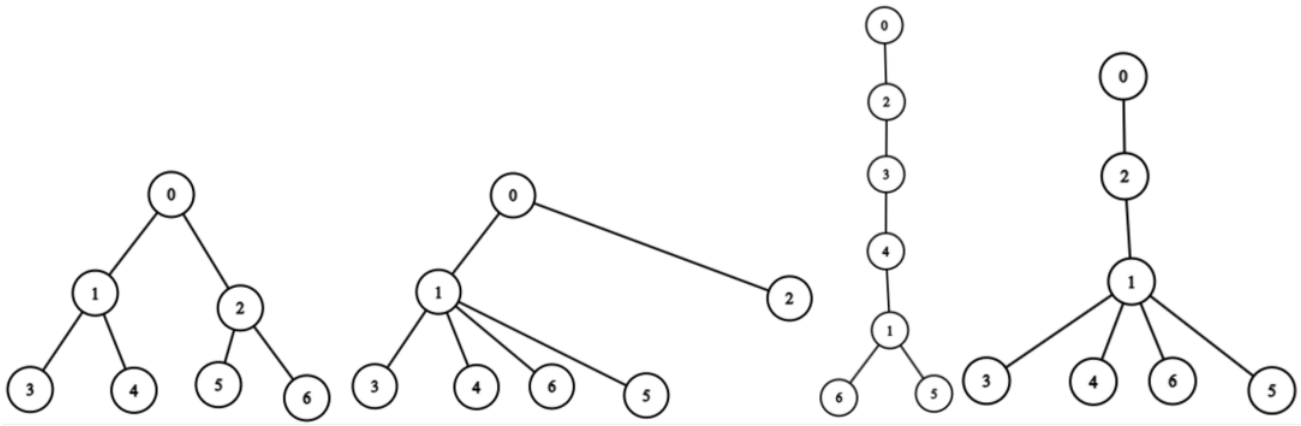

Figure 10. Different topologies with the same number of nodes.

The following calculation reward is for the graph in Figure 11.

$$\text{Weight}(\text{Node}_0) = 7 \rightarrow \text{WeightN}(\text{Node}_0) = \frac{7}{7+2+3+1} = \frac{7}{13} \quad (42)$$

$$\text{Weight}(\text{Node}_1) = 2 \rightarrow \text{WeightN}(\text{Node}_1) = \frac{2}{7+2+3+1} = \frac{2}{13} \quad (43)$$

$$\text{Weight}(\text{Node}_2) = 3 \rightarrow \text{WeightN}(\text{Node}_2) = \frac{3}{7+2+3+1} = \frac{3}{13} \quad (44)$$

$$\text{Weight}(\text{Node}_6) = 1 \rightarrow \text{WeightN}(\text{Node}_6) = \frac{1}{7+2+3+1} = \frac{1}{13} \quad (45)$$

$$\text{Weight}(\text{Node}_3) = \text{Weight}(\text{Node}_4) = \text{Weight}(\text{Node}_5) = \text{Weight}(\text{Node}_7) = 0 \quad (46)$$

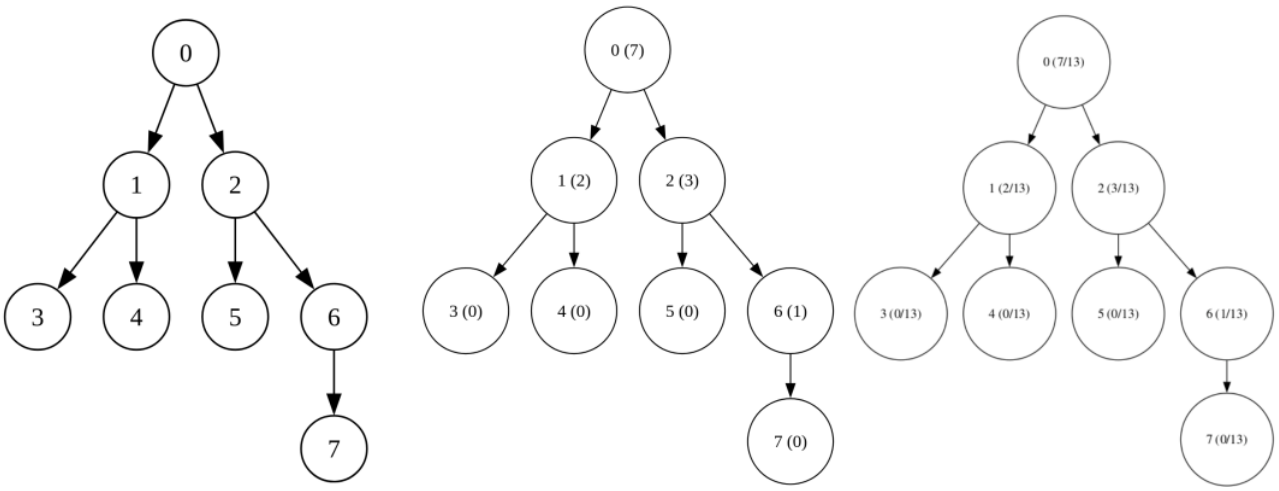

Figure 11. A step-by-step for calculating the reward of each node.

In general, for each node, we have the following formulas:

$$\text{WeightN}(\text{Node}_i) = \frac{\text{Weight}(\text{Node}_i)}{\sum_{j=0}^{N} \text{Weight}(\text{Node}_j)} \quad (47)$$

where the $\text{Weight}(\text{Node}_i)$ can be calculated using the following recursive formula with $C_{i,j}$ is the coefficient of the graph matrix, meaning $C_{i,j}$ will be equal to 0 if Node_i and Node_j do not have any connection.

$$\text{Weight}(\text{Node}_i) = \sum_{j=0}^{N} C_{i,j}\big(\text{Weight}\big(\text{Node}_j\big) + 1\big) \tag{48}$$

For example, if we want to calculate the $\text{Weight}(\text{Node}_1)$ using our formula:

$$\text{Weight}(\text{Node}_1) = \sum_{j=0}^{N} C_{1,j}\big(\text{Weight}\big(\text{Node}_j\big) + 1\big)$$

Since Node_1 has communication only with Node_3 and Node_4,we will have:
$C_{1,0} = C_{1,1} = C_{1,2} = C_{1,5} = C_{1,6} = C_{1,7} = 0 \text{and} C_{1,3} = C_{1,4} = 1$
Thus, $\text{Weight}(\text{Node}_1) = (\text{Weight}(\text{Node}_3) + 1) + (\text{Weight}(\text{Node}_4) + 1)$
And, since Node_3 and Node_4 do not have any connection/sub-children for any j, we will have $C_{3,j} = C_{4,j} = 0$,therefore:
$\text{Weight}(\text{Node}_1) = (0 + 1) + (0 + 1) = 2$

The reward system design considers the potential of nodes to try to game the system. While nodes may attempt to increase their connections or manipulate their rank artificially, the system relies on a combination of the trust matrix and application of depth-first search (DFS) for weight calculation. This ensures that the connections' quantity, quality, and authenticity are considered. Nodes with superficial connections that do not genuinely contribute to network health may receive insignificant rewards. Additionally, the continuous recalibration of the trust coefficients serves as a check against nodes that may attempt to manipulate their standing. Malicious or manipulative actions can lead to a decrease in the trust coefficients, rendering such strategies counterproductive.

7. Actions Based on The Trust Matrix

Each node will have an objective to maximize its rewards; therefore, it aims to optimize the function $\text{maxWeightN}(\text{Node}_i)$ because the sum of all weights is fixed. The node is looking more to optimize its weight, thereby maximizing connections with nodes that have larger weights. To avoid greedy nodes and conflicts between nodes, in this model, a node should also consider maximizing the collective gain of the group because we have the same reward for each transaction, and we can focus more on increasing the depth of the reward tree. Therefore, the node should solve the following optimization problem.

$$\begin{cases} \text{maxWeight}(\text{Node}_i) \\ \max \sum_{i=0}^{i=N} \min(\cos(2\pi \text{Weight}(\text{Node}_i)), \text{Weight}(\text{Node}_i)) \end{cases} \tag{49}$$

The second constraint involves simply calculating the number of nodes with a weight different than zero; the larger this number, the larger the depth of the reward tree. To have a formal equation to count the number of non-zero weight nodes, the first thing is to omit the difference between weights that are different than 0; the function $\cos(2\pi x)$ for all integers has this property. Therefore, any integer number will result as an output of 1, however, $\cos(2\pi x)$ is not sufficient since we will have $\cos(0) = 1$. Therefore, the trick is to calculate the $\min(x, \cos(2\pi x))$; by having this function, we extract all the weights different than 0 and sum them up. Each node, as mentioned previously, has its trust matrix and can decide based on that, by either removing the connection or trying to add a connection. We can define two coefficients, coefficient of lost and coefficient of trust. If the value in that matrix is below the coefficient of lost, the node can remove its connection from that node; if the value is greater than the coefficient of trust, it can try to reach that node and make a connection with it. These values can affect the matrix's convergence speed in addition to the two coefficients of reward and punishment.

Coefficient of Lost: This coefficient is crucial for penalizing malicious or non-cooperative behavior in the network. A higher value would mean that trust is lost rapidly after a few malicious actions, making the network more stringent but potentially more volatile, as nodes might be penalized too harshly for occasional non-malicious lapses. A lower value would make the network more forgiving and expose it to risks if malicious nodes are not adequately penalized.

Coefficient of Trust: This coefficient determines how quickly trust can be gained within a network. A higher value would allow nodes to quickly regain trust after demonstrating cooperative behavior, making the network more adaptive and resilient to occasional honest mistakes. However, if set too high, malicious nodes may quickly regain trust after a few honest actions, which can be exploited later. A lower value ensures trust is gained slowly, making the network more stable in its trust evaluations and potentially more resistant to acknowledging reformative behavior.

The interplay between the coefficients and their values is crucial. Ideally, they should be set such that the network is resilient to occasional lapses but stringent against consistently malicious behavior. Too high values for both could make the network volatile, whereas too low values might make the network sluggish and less adaptive.

8. Analysis and Evaluation of Results

In our research, we aimed to combine graph theory and game theory principles to enhance the structure of decentralized blockchain networks. To test our theoretical ideas and better understand the practical effects of our algorithm, we conducted a series of comprehensive simulations. Below, we provide a detailed overview of our simulation setup:

- Types of Networks: Our study explored various types of networks, including scale-free, small-world, and random graph networks. By examining this range, we ensure that our findings are relevant to multiple network models.
- Network Parameters: To probe the adaptability of the algorithm across varying scales, our simulations spanned networks with 10, 100, 1,000, and 10,000 nodes. This staggered approach offers a multi-dimensional perspective, shedding light on scalability and performance issues.
- Simulation Environment: Python was used for our simulations because it has a wide range of science-related tools and is very flexible. Its strong reputation in research has helped us obtain reliable and repeatable results.
- Diverse Datasets and Parameter Tuning: Our simulations engaged varied datasets, manipulated the malicious probability of nodes, toggled cooperative and non-cooperative strategies, and fine-tuned the reward and punishment coefficients. This dynamic landscape enriches our results, revealing the agility of the algorithm across different scenarios. Simulation Algorithms: We detail the algorithms employed in our simulations below to provide a more granular perspective on our methodology.

The process of updating the trust matrix in our system is a crucial aspect of maintaining network integrity and efficiency. The algorithm we developed for this purpose systematically adjusts the trust scores based on the latest interactions and transaction verifications. Algorithm 1 details the steps involved in this update process, demonstrating how the trust matrix is modified in response to the nodes' behaviors. The algorithm iteratively evaluates and updates the trust scores between pairs of nodes based on their recent activities, ensuring that the matrix accurately reflects the current state of trust within the network.

The reward system in our network is designed to incentivize nodes for their participation and contributions. It operates by computing rewards for each node based on specific criteria, such as the number of sub-children a node has or its role in the network. To systematically calculate these rewards, we developed 'Algorithm 2'. This algorithm outlines the process for determining the reward each node receives, based on its position and function within the network. It takes into account various factors, ensuring a fair and equitable distribution of the network rewards among all participating nodes.

Algorithm 1. Algorithm for updating the trust matrix.

```
Function updateTrustMatrix(node):
    For i from 0 to nodesCount:
        For j from 0 to nodesCount:
            If node.truthMatrix[i][j] == 1:
                node.trustMatrix[i][j] = (node.trustMatrix[i][j]+1)/2
            Else If node.truthMatrix[i][j] == -1:
                node.trustMatrix[i][j] = node.trustMatrix[i][j]/2
            endIf
        end-For
    end-For
end
```

Algorithm 2. Algorithm for the reward system

```
Function computeReward(node,networkReward):
    subChildrenCount = SDF(node,networkReward)
end
```

In our network, nodes actively make decisions based on the trust levels established within their trust matrices. These decisions involve actions like removing less trusted neighbors or adding new ones, which are crucial for maintaining the network's integrity and efficiency. To systematically facilitate these actions, we developed 'Algorithm 3'. This algorithm outlines the procedural steps each node follows when performing actions based on their trust assessments. It ensures that nodes dynamically adjust their connections to optimize the network's trustworthiness and resilience.

Algorithm 3. Algorithm for node actions

```
Function performActions(node,network):
    If average(node.trustMatrix[node.index] < THRESHOLD:
        removeLeastTrustedNeighbor(node)
    Else If average(node.trustMatrix[node.index]) > HIGHTRUST:
        addNewNeighbor(node, network)
    endIf
end
Function removeLeastTrustedNeighbor (node):
    leastTrustedNeighbor = argmin(node.trustMatrix[node.index])
    REMOVE node.neighbors[leastTrustedNeighbor]
end
Function addNewNeighbor(node, network):
    potentialNeighbors = SetDifference (network.nodes, node.neighbors)
    newNeighbor = randomChoice(potentialNeighbors)
    Append newNeighbor To node.neighbors
end
```

Now that we have set up the scene in more detail, let us move on to detailed and interesting discoveries from our study.

1. Packet Loss vs. Malicious Nodes

The impact of malicious nodes on network functionality is a pressing concern in blockchain systems. Our model demonstrates a linear relationship between the percentage of malicious nodes and the packet loss. However, a noticeable trend was the exponential increase in packet loss when the malicious nodes exceeded a certain threshold, which was

linked to the inherent limitations of the PBFT algorithm. Specifically, when malicious nodes approach a threshold, packet losses surge, highlighting the importance of maintaining node integrity in the network. Figure 12 graphically represents this relationship, showing the correlation between packet loss percentage and the percentage of malicious nodes in the system.

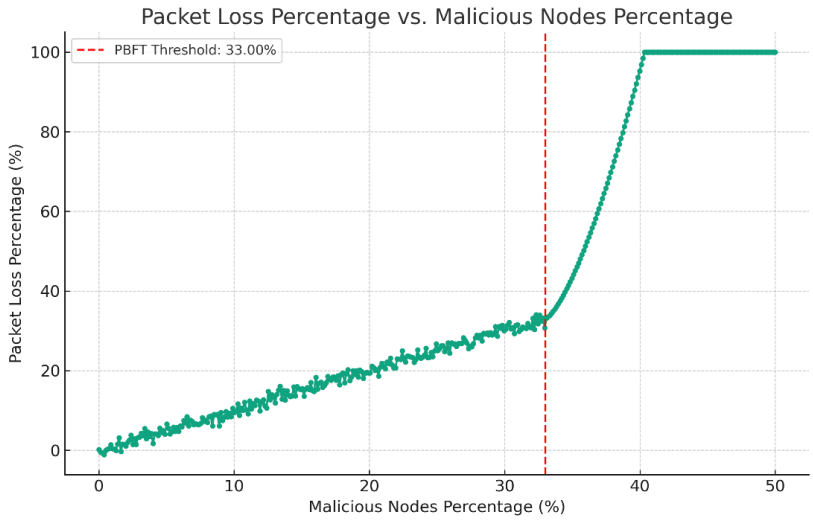

Figure 12. Packet loss percentage vs. malicious nodes percentage.

2. False Positive/Negative Rate Over Rounds

Consistency and accuracy are of paramount importance in decentralized systems. Our model exhibited a declining trend in false-positive and false-negative rates as the rounds advanced. This is attributed to our game-theoretic framework's refined matrix representation and evolving nature, which iteratively improves its discernment capabilities, ensuring reduced erroneous detections in each subsequent round. Figure 13 illustrates this trend, providing a graphical representation of how the false positive/negative rates change over successive rounds.

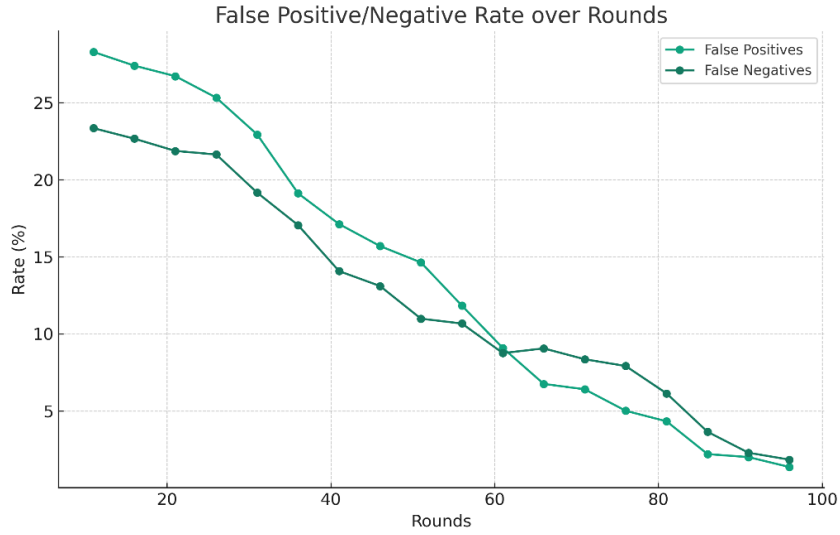

Figure 13. False positive/negative rate over rounds.

3. Resilience to Sybil Attacks Over Rounds

The mettle of a blockchain network is also judged by its resilience to Sybil attacks. With each progressive round, the number of successful attacks decreased, emphasizing the effectiveness of our matrix-based representation and reward mechanism. Thus, the network exhibits enhanced defensive capabilities against adversarial strategies. Figure 14 visually demonstrates this trend, showcasing the resilience of the network to Sybil attacks over successive rounds and highlighting the declining frequency of successful attacks as the rounds progress.

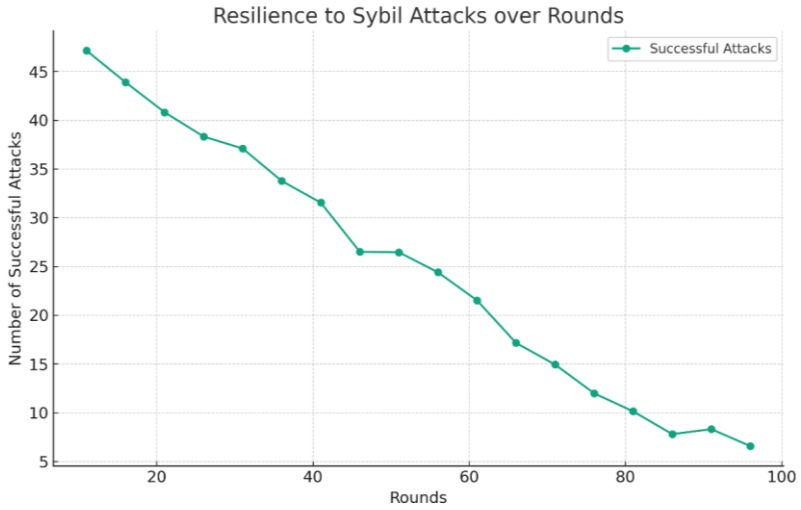

Figure 14. Resilience to Sybil attacks over rounds.

4. Convergence Time Over Rounds

The matrices should converge by leveraging the theoretical premise stipulating that given non-malicious nodes operate under a unified strategy. Our simulations mirrored this assumption. The convergence time, indicative of the disparity in trust matrices, exhibited a decreasing trajectory over the rounds. This faster consensus achievement underscores the network's increasing efficiency, ensuring nodes reach an agreement more promptly than transactions and data validation. Figure 15 graphically illustrates this trend, showing the convergence time over successive rounds and highlighting the network's enhanced efficiency in reaching consensus as the rounds progress.

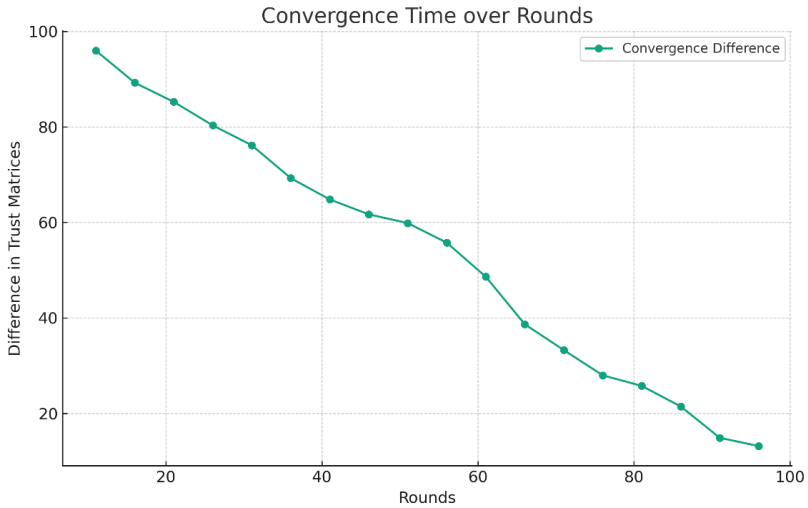

Figure 15. Convergence time over rounds.

In summary, the proposed incentive model's combination of graph and game theory offers a novel approach to fortifying blockchain security. Matrix representation, integrated with a reward mechanism and governed by the principles of Nash Equilibrium, heralds a new era of decentralized systems that are secure and efficient. The simulations represent a testament to the model's prowess in combating malicious endeavors, thus bolstering the integrity and trustworthiness of the blockchain network.

9. Conclusions

Blockchain technology's potential to revolutionize various sectors requires robust and efficient incentive mechanisms to maintain network integrity and participation. This study delved into the intricate landscape of blockchain networks, highlighting the importance of node incentivization and trust optimization. Via a comprehensive review of existing studies, we identify gaps and potential areas of improvement in current incentive models.

Our proposed framework, which synergizes graph and game theories, offers a novel approach to fostering node cooperation. Simulations conducted across diverse network typologies demonstrate the efficacy and scalability of our model. Our approach ensures a more democratized network ecosystem and paves the way for greater decentralization and network security.

As promising as our findings are, blockchain technology's dynamic and evolving nature demands continuous research and development. Therefore, we envision several compelling directions for future research.

1. Enhancing Privacy with Zero-Knowledge Proofs: ZKP can enable nodes to verify transactions or interactions without revealing underlying data in the trust matrix. This approach ensures that the integrity and confidentiality of the trust matrix are maintained even when nodes are required to share or prove certain information. Research in this direction should focus on developing ZKP protocols tailored to the specific requirements of our framework, ensuring that they are efficient and scalable.
2. Secure Multi-Party Computation for Decentralized Calculations: MPC allows for the computation of trust matrix updates and decisions in a decentralized and secure manner, where no single node can access the complete data. This method is particularly beneficial in a blockchain environment because it aligns with the principles of decentralization and mutual distrust. Future research could explore how MPC can be effectively integrated into the blockchain network to compute outcomes securely based on the trust matrix while ensuring that the individual data of each node remain private.
3. Cross-Chain Compatibility and Interoperability: Another vital area for future exploration is the adaptability of the incentive model to different blockchain platforms. Cross-chain compatibility and interoperability are crucial to achieving a more interconnected and versatile blockchain ecosystem. This research involves creating standardized protocols or frameworks that enable seamless interactions and integration between various blockchain networks, thus enhancing our model's overall utility and reach.

As blockchain continues its trajectory toward global adoption, the optimization of node incentives remains paramount. This study contributes a significant step in this direction by offering insights and methodologies that could shape the future of decentralized networks.

Author Contributions: Conceptualization, S.M. and A.A.E.K.; methodology, S.M.; software, S.M.; validation, S.M. and A.A.E.K.; formal analysis, S.M.; investigation, S.M.; resources, S.M.; data curation, S.M.; writing—original draft preparation, S.M.; writing—review and editing, S.M. and A.A.E.K.; visualization, S.M.; supervision, A.A.E.K.; project administration, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sanka, A.I.; Irfan, M.; Huang, I.; Cheung, R.C. A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research. *Comput. Commun.* **2021**, *169*, 179–201. [CrossRef]
2. Yang, S.; Zhang, F.; Huang, K.; Chen, X.; Yang, Y.; Zhu, F. Sok: Mev countermeasures: Theory and practice. *arXiv* **2022**, arXiv:2212.05111.
3. Stiglitz, J.E. Pareto Optimality and Competition. *J. Finance* **1981**, *36*, 235–251. [CrossRef]
4. Nakamoto, S.; Bitcoin: A Peer-to-Peer Electronic Cash System. December 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 21 August 2023).
5. Antonopoulos, A.M. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*; O'Reilly Media, Inc.: Newton, MA, USA, 2014.
6. Mingxiao, D.; Xiaofeng, M.; Zhe, Z.; Xiangwei, W.; Qijun, C. A review on consensus algorithm of blockchain. In Proceedings of the 2017 IEEE International Conference on Systems, and Cybernetics (SMC), Man, Banff, AB, Canada, 5–8 October 2017; pp. 2567–2572.
7. Osborne, M.J. *An Introduction to Game Theory*; Oxford University Press: Oxford, UK, 2004; Volume 3.
8. Fudenberg, D.; Tirole, J. *Game Theory*; MIT press: Cambridge, MA, USA, 1991.
9. Liu, Z.; Luong, N.C.; Wang, W.; Niyato, D.; Wang, P.; Liang, Y.-C.; Kim, D.I. A survey on applications of game theory in blockchain. *arXiv* **2019**, arXiv:1902.10865.
10. Kreps, D.M. Nash equilibrium. In *Game Theory*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 167–177.
11. Li, Z.; Shen, H. Game-Theoretic Analysis of Cooperation Incentive Strategies in Mobile Ad Hoc Networks. *IEEE Trans. Mob. Comput.* **2011**, *11*, 1287–1303. [CrossRef]
12. Mahmoud, M.M.E.A.; Shen, X. FESCIM: Fair, Efficient, and Secure Cooperation Incentive Mechanism for Multihop Cellular Networks. *IEEE Trans. Mob. Comput.* **2011**, *11*, 753–766. [CrossRef]
13. Dias, J.A.; Rodrigues, J.J.; Shu, L.; Ullah, S. Performance evaluation of a cooperative reputation system for vehicular delay-tolerant networks. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 1–13. [CrossRef]
14. Yang, G.; He, S.; Shi, Z.; Chen, J. Promoting Cooperation by the Social Incentive Mechanism in Mobile Crowdsensing. *IEEE Commun. Mag.* **2017**, *55*, 86–92. [CrossRef]
15. He, Y.; Li, H.; Cheng, X.; Liu, Y.; Yang, C.; Sun, L. A Blockchain Based Truthful Incentive Mechanism for Distributed P2P Applications. *IEEE Access* **2018**, *6*, 27324–27335. [CrossRef]
16. Buterin, V. Ethereum Whitepaper. Available online: https://ethereum.org/en/whitepaper/ (accessed on 21 August 2023).
17. Han, R.; Yan, Z.; Liang, X.; Yang, L.T. How Can Incentive Mechanisms and Blockchain Benefit with Each Other? A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–38. [CrossRef]
18. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the Security and Performance of Proof of Work Blockchains. In Proceedings of the CCS'16: 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria, 24–28 October 2016; pp. 3–16.
19. King, S.; Nadal, S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Available online: https://decred.org/research/king2012.pdf (accessed on 19 November 2023).
20. Alzahrani, N.; Bulusu, N. Towards True Decentralization: A Blockchain Consensus Protocol Based on Game Theory and Randomness. In *Decision and Game Theory for Security*; GameSec 2018; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; pp. 465–485.
21. Yun, J.; Goh, Y.; Chung, J.-M. Trust-Based Shard Distribution Scheme for Fault-Tolerant Shard Blockchain Networks. *IEEE Access* **2019**, *7*, 135164–135175. [CrossRef]
22. Huang, C.; Wang, Z.; Chen, H.; Hu, Q.; Zhang, Q.; Wang, W.; Guan, X. RepChain: A Reputation-Based Secure, Fast, and High Incentive Blockchain System via Sharding. *IEEE Int. Things J.* **2021**, *8*, 4291–4304. [CrossRef]
23. Qiu, X.; Qin, Z.; Wan, W.; Zhang, J.; Guo, J.; Zhang, S.; Xia, J. A Dynamic Reputation–based Consensus Mechanism for Blockchain. *Comput. Mater. Contin.* **2022**, *73*, 2577–2589. [CrossRef]
24. Li, X.; Liu, Q.; Wu, S.; Cao, Z.; Bai, Q. Game theory based compatible incentive mechanism design for non-cryptocurrency blockchain systems. *J. Ind. Inf. Integr.* **2023**, *31*. [CrossRef]
25. Liu, X.; Huang, Z.; Wang, Q.; Wan, B. An Evolutionary Game Theory-Based Method to Mitigate Block Withholding Attack in Blockchain System. *Electronics* **2023**, *12*, 2808. [CrossRef]
26. Windiatmaja, J.H.; Hanggoro, D.; Salman, M.; Sari, R.F. PoIR: A Node Selection Mechanism in Reputation-Based Blockchain Consensus Using Bidirectional LSTM Regression Model. *Comput. Mater. Contin.* **2023**, *77*, 2309–2339. [CrossRef]
27. West, D.B. *Introduction to Graph Theory*; University of Illinois: Upper Saddle River, NJ, USA, 2001; Volume 2.
28. Gilles, R.P. *The Cooperative Game Theory of Networks and Hierarchies*; Springer Science and Business Media LLC: Dordrecht, The Netherlands, 2010.

29. Liu, X.; Zhang, J.; Zhu, P. Modeling cyber-physical attacks based on probabilistic colored Petri nets and mixed-strategy game theory. *Int. J. Crit. Infrastruct. Prot.* **2017**, *16*, 13–25. [CrossRef]
30. Negy, A.N.; Rizun, P.R.; Sirer, E.G. Selfish mining re-examined. In *International Conference on Financial Cryptography and Data Security*; Springer: Cham, Switzerland, 2020.
31. Bag, S.; Ruj, S.; Sakurai, K. Bitcoin Block Withholding Attack: Analysis and Mitigation. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 1967–1978. [CrossRef]
32. Saad, M.; Njilla, L.; Kamhoua, C.; Mohaisen, A. Countering selfish mining in blockchains. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019.
33. Sukhwani, H.; Martinez, J.M.; Chang, X.; Trivedi, K.S.; Rindos, A. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In Proceedings of the IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, 26–29 September 2017; pp. 253–255.
34. Tarjan, R. Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1972**, *1*, 146–160. [CrossRef]