I. J. Computer Network and Information Security, **2024, 2, 92-102**
Published Online on April 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.02.08

An Efficient and Secure Blockchain Consensus Algorithm Using Game Theory

Naveen Arali
KLE Technological University, School of Computer Science and Engineering, Hubli, 580031, India
E-mail: naveenarali01@gmail.com

Narayan D. G.*
KLE Technological University, School of Computer Science and Engineering, Hubli, 580031, India
E-mail: narayan_dg@kletech.ac.in
ORCID iD: https://orcid.org/0000-0002-2843-8931
*Corresponding Author

Altaf Husain M.
KLE Technological University, School of Computer Science and Engineering, Hubli, 580031, India
E-mail: altaf.husain@kletech.ac.in

P. S. Hiremath
KLE Technological University, School of Computer Science and Engineering, Hubli, 580031, India
E-mail: pshiremath@kletech.ac.in

Received: 08 July 2023; Revised: 16 October 2023; Accepted: 08 November 2023; Published: 08 April 2024

Abstract: Blockchain technology is a decentralized ledger system that finds applications in various domains such as banking, e-governance, and supply chain management. The consensus algorithm plays a crucial role in any blockchain network as it directly impacts the network's performance and security. There have been several proposed consensus mechanisms in the literature, including Proof of Work (PoW), Proof of Stake (PoS), Robust Proof of Stake (RPoS), and Delegated Proof of Stake (DPoS). Both Ethereum and Bitcoin utilize the PoW consensus mechanism, where nodes compete to solve puzzles in order to generate blocks, consuming significant processing power. On the other hand, the PoS consensus mechanism selects miners based on the stakes they hold, making it more energy efficient. However, PoS has drawbacks such as vulnerability to coin age accumulation attacks and the potential for partial centralization. In this work, we present a consensus mechanism known as Delegated Proof of Stake with Downgrading Mechanism using Game Theory (DDPoS (GT)). This mechanism employs a two-step game strategy to divide nodes into strong and weak nodes, as well as attack and non-attack nodes. Later, the results of the two games are combined to enhance protocol efficiency and security. Experimental results using a private Ethereum-based network demonstrate that DDPoS (GT) performs better than PoS and DPoS in terms of transaction latency, average block waiting time, and fairness.

Index Terms: Blockchain, Consensus, POW, POS, DPOS, Ethereum, Game Theory.

1. Introduction

Blockchain is a rapidly emerging technology characterized by a continually expanding series of interconnected records, known as blocks, which are secured using cryptographic principles. While initially developed for cryptocurrencies like Bitcoin, blockchain has since extended its applications to various fields. Each block in the blockchain contains a cryptographic hash of the preceding block, along with a timestamp and transaction information. The immutability of the data ensures its resistance to easy modification. Functioning as an open and distributed ledger, blockchain facilitates efficient and secure recording of transactions between parties in a peer-to-peer network. The transparency and verifiability of transactions are achieved by eliminating the need for intermediaries. Within the blockchain, anyone can access the transaction history, observing previously completed transactions. This decentralized approach employs cryptographic hash functions to establish links between blocks, while a Merkle tree is used to maintain transaction integrity.

Because blockchain operates in a decentralized manner, it lacks a centralized authority to govern it reliably. Anyone who wishes to participate can join the network. The absence of a trustworthy central authority means that any individual can act as a node, and all nodes within the network are considered untrustworthy. Therefore, a consensus mechanism is necessary for all nodes to agree on the network's state. The consensus method determines the authenticity of transactions and dictates the behavior of each node. The consensus protocol influences the performance of blockchain significantly [1]. The consensus mechanism employed in a blockchain technology determines factors such as efficiency, fairness, security, and integrity. Numerous consensus algorithms have been introduced in the literature and they can be classified into various categories, including puzzle-based, stake-based, Byzantine Fault Tolerant (BFT)-based, hybrid, and more. In real-time blockchain networks such as Bitcoin and Hyperledger Fabric, consensus algorithms like Proof of Work (PoW) and Reliable, Replicated, Redundant, and Fault-Tolerant (RAFT) are commonly employed.

Ethereum 2.0 has adopted the Proof of Stake (PoS) consensus mechanism replacing the earlier Proof of Work (PoW) algorithm. PoS offers energy efficiency and reduces the environmental impact compared to PoW by not requiring miners to perform complex computations. Additionally, PoS encourages long-term network participation and security by rewarding users who "stake" their cryptocurrency holdings. Various PoS variations, including Proof of Authority (PoA) and Delegated Proof of Stake (DPoS), have been proposed in the literature. Nevertheless, the necessity for continuous improvements in efficiency, fairness, and security within the blockchain systems remains an important subject of ongoing research [2]. Additionally, as most of the work in the literature focus on design and evaluation of new consensus algorithms based on POW or POS through simulations, there is a need for design and experimental evaluation of new consensus algorithm in real-time blockchain platforms. To the best of our knowledge, there exist limited instances of consensus design, such as the one proposed in [3], that have been implemented in real-time utilizing the Geth code of Ethereum.

In this work, we present a new consensus mechanism known as Delegated Proof of Stake with Downgrading Mechanism using Game Theory, abbreviated as DDPoS (GT), which combines delegated proof of stake consensus mechanism with game theory. We employ a three-stage game to classify nodes as either strong or weak, and determine whether they engage in attack or non-attack behaviors. The outcomes of the initial two games are then leveraged in the third stage to enhance the efficiency and security of the protocol. Furthermore, we design and evaluate the proposed work in private Ethereum-based network. The contributions of this work are as follows:

- Designed an efficient and secure consensus algorithm based on DPoS using Game Theory.
- Implemented the proposed algorithm in a Ethereum-based multi-node blockchain network.
- Evaluated the proposed algorithm with POS and DPOS algorithms in terms of different performance metrics using various scenarios.

The rest of this paper is organized as follows. In Section 2, we discuss the related work on consensus algorithm design focusing on game theory. Section 3 describes the proposed methodology and the algorithms used in the implementation. Section 4 presents the results of the proposed system on experimental real-time Ethereum network. Finally, we discuss conclusions and future work in Section 5.

2. Related Works

Satoshi Nakamato introduced the concept of Bitcoin, which is a decentralized digital currency enabling direct peer-to-peer electronic cash transactions without the need for intermediaries such as banks. Author proposed the Proof of Work (PoW) consensus algorithm as a means to establish an electronic transaction system that operates without reliance on trust. Numerous advancements have been proposed in the literature concerning Proof of Work (PoW) algorithms. In [4], the authors conduct a mathematical analysis aimed at identifying weaknesses in the blockchain. The study takes into account parameters such as transactions per second, mining difficulty, the number of miners in the network, and the hash rate. One limitation observed is that increasing hash calculations lead to longer expected mining times. In [5], the authors propose predicting the real-time total hash rate to maintain a stable block creation time. They examine parameters such as hash rate and mining difficulty adjustment. However, one limitation discovered is that the difficulty adjustment algorithm results in longer block creation times at certain intervals under simulated conditions. To address these challenges, the authors simulate a real-time Proof of Work consensus algorithm using a network model that considers high hash rate fluctuations. The simulation is implemented using the Python programming language.

In [6], the authors investigate a two-layer neural network algorithm designed to regulate block difficulty. The study considers parameters such as fast updating, low volatility, and hash rate. However, a limitation is noted: the overall accuracy of the neural network falls below 90 percent. The authors conduct a Monte-Carlo simulation of the algorithm, utilizing real data from Ethereum. In [7], the authors focus on block compression using optimization and block compression models to reduce block size. The parameters examined include transmission efficiency, storage space, mining difficulty, transaction count, energy consumption, and security. A limitation of their work is that they utilize a single data compression algorithm. It is suggested that block compression could be applied to other consensus protocols in future research. The authors implement a prototype using the Go language, with the deflate algorithm employed for compressing or decompressing block data. The issue of high energy consumption in Proof of Work (PoW) has led to a

growing trend in the development of stake-based consensus mechanisms. In [8], the authors introduce the first formal economic model of Proof of Stake (PoS). They establish certain conditions using mathematical models, such as probabilities, to analyze how PoS generates consensus. In PoS, the selection of the miner is based on the maximum stakes held by a node. Meanwhile, in [9], the focus is on enhancing the energy efficiency of cryptocurrencies. The authors propose three potential scenarios for transitioning from PoW to PoS. They conclude that deploying a new consensus algorithm is necessary, but the adoption of a new mechanism like PoS requires the majority agreement of existing nodes. In [10], the authors propose the Robust Proof of Stake (PoS) consensus algorithm, which utilizes the age of coins instead of the number of coins for miner selection. This approach aims to reduce the vulnerability to coinage accumulation attacks, a concern in traditional PoS systems. The parameter of coinage over the number of coins is employed to prioritize older nodes in the network, rather than solely relying on the quantity of coins held by a node. In [11], authors propose DPOS mechanism with downgrading mechanism. However, authors use simulation study to evaluate the proposed work. In [12], the authors propose an optimization algorithm to address shortcomings of DPOS. Authors use Borda Count to compute preference score statistics on candidate nodes. In [13], the authors propose algorithm namely modified Proof-of-Probability (PoP). In this work, the PoP nodes perform a modulo operation on the nonce value, which is then compared with the expected value provided by the super node selected by the DPoS nodes.

In [14], the authors introduce BAZO, a POS based blockchain. BAZO enhances the degree of randomness in the selection of the next validator in PoS consensus mechanisms at each block height. By incorporating transaction aggregation and double-linked blocks, BAZO improves scalability. Evaluations of the BAZO blockchain demonstrate its effectiveness in mitigating attacks such as the 51\% attack, double spending, and grinding attacks while avoiding centralization. In [15], the authors discuss the applicability of PoS in permissionless blockchain platforms but highlight the security shortcomings of existing PoS variants. They address issues related to nothing-at-stake, long-range attacks, and stake grinding attacks, which can significantly compromise blockchain security. To overcome these problems, they propose a secure Proof of Stake protocol, PoTS, that leverages Trusted Execution Environments (TEEs). This protocol resolves the nothing-at-stake problem and a large class of long-range attacks, with the combination of TEEs enhancing security. In [16], the authors propose Proof of Game (PoG) consensus algorithm that can be applied to both single-player and multiplayer challenges. The authors also highlight that the presence of selfish miners can lead to longer block waiting time as difficulty increases. In [17], the authors propose a bi-level optimization model based on the Stackelberg game. The proposed approach considers equilibrium strategies including moderating and compensating tactics. The optimization model using adaptive differential evolution is used. In [18], the authors introduce the Proof of Activity (PoA) protocol, which incorporates game theory. POA defends against majority attacks and selfish mining while consuming minimal energy. The PoA protocol aims to enhance the security and energy efficiency of the consensus mechanism in blockchain systems.

In [19], the authors adopt learning game theory to simplify and prove convergence in problem-solving. They emphasize the resilience and autonomy gained by the algorithm through learning behavior. The suggested approach offers a fresh perspective on examining consensus challenges. In [20], the authors introduce a new approach where transactions are divided among multiple shards and processed concurrently. They propose a two-phase bargaining game model that dynamically adapts to the state of the blockchain network, providing a strategic solution to the shard-based consensus challenge. They also discuss the integration of blockchain with other technologies, present their findings, and recommend important research topics. In [21], the authors propose a POS based consensus mechanism using game theory. Authors use federated learning to compute the trust scores. Furthermore, authors use a game-theory for reward and punishment during the mining process.

Based on the literature review, Table 1 gives the summary of popular consensus algorithms and their features. The table shows that each algorithm has its own advantages and disadvantages. Furthermore, most of the works carried out in the above literature focus on implementing and evaluating consensus algorithms through simulations. In this work, we propose a new consensus mechanism called DDPoS (GT) for real-time Ethereum-based network by using game theory to enhance fairness among nodes in the network.

Table 1. Research gaps identified

Property	Consensus Algorithms		
	PoW	PoS	DPoS
Blockchain type	Permissionless	Both	Both
Concept	Computational puzzle	Percentage of stakes	Delegates and Stake
Efficiency	Low	Medium	High
Security	High	Low	Medium
Fairness	High	Low	Medium

3. Proposed Methodology

This section describes the delegated proof of stake system with a downgrading mechanism utilizing game theory and the implementation algorithm.

3.1. System Model

The proposed system design of the game theory mechanism for DDPoS (GT) is described in this section. The basic phase in implementing any consensus mechanism is to mine the block, which entails establishing a valid hash using a nonce value and the previous block's hash value. After that, make a new block. The flow of three-stage games is depicted in Fig. 1. Based on their outgoing sake, the first stage game determines whether the nodes will choose strong pool or weak pool. The nodes select whether to attack or not attack in the second stage game, while the final stage game is based on the previous two stages. After the three stages are completed, the system selects the best miner for the network. For validating all of the transactions and mining the block, each miner receives a reward. To obtain an agreement between nodes, the consensus selects nodes from the strong pool network. Only a few nodes from the strong pool are chosen by the algorithm, and they have the authority to create and mine blocks in the blockchain network. Because there may be numerous malicious nodes in the network, we degrade in the second stage game to overcome this Byzantine fault.

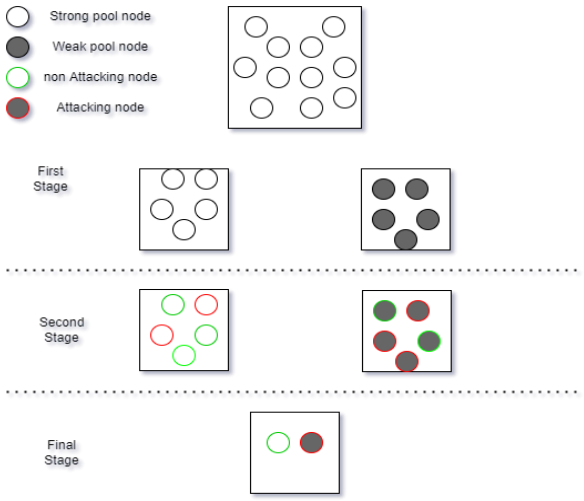

Fig.1. System model

3.2. Algorithms

In this section, we discuss steps involved in proposed work along with the algorithms used in the implementation.

First Stage Game

The Algorithm 1 illustrates node strategies, such as whether they will choose the strong or weak pool. Let $N = \{n_1, n_2, \cdots, n_i\}$, where N is the number of nodes in the network. Let $S_i = s_1, s_2, \cdots, s_n$, where S_i denotes the stakes of each of the n nodes. AS represents the network's average stake. The average Stake will be calculated as follows.

$$AS = \frac{\sum S_i}{N} \tag{1}$$

Strong pool will be chosen by the node with a stake bigger than the network's average stake $(S_i > AS)$. The node with a stake that is lower than the network's average will choose the weak pool $(S_i < AS)$. After selecting a pool, both pools will sort their nodes according to stakes. The matrix of two players choosing the pools is shown in Table 2. Let l represent the loss incurred as a result of selecting attack. Let ls represent degraded stakes. The matrix depicts the loss of stakes.

Table 2. Payoff matrix of first stage game

	WeakPool	StrongPool
Weak pool	$l = S_i - ls,\ l = S_i - ls$	$l = S_i - ls,\quad 0$
StrongPool	$0,\ l = S_i - ls$	$0,\quad 0$

- when both players are part of the Strong Pool, the mining capacity of Strong Pool nodes consistently exceeds that of Weak Pool nodes. This eliminates the risk of potential attacks, ensuring zero losses. Let MP_i denotes the N_i node's mining power were

$$MP_i > avgMiningpower \tag{2}$$

- when one player comes from the strong pool and the other from the weak pool. The mining power of the weak

pool node is always lower than the mining power of the strong pool node. The weak pool node has the option of attacking.

$$ls = \frac{N_i\,(25 \cdot S_i)}{100} \tag{3}$$

$$l = S_i - ls \tag{4}$$

- when both nodes are from the same weak pool. The mining power of both nodes is less than the average mining power. If the mining power is larger than 1/4th mining power, then the node will try to attack.

$$\frac{1}{4} < MP_i < avgMiningpower \tag{5}$$

Algorithm 1 First Stage Game

Require: Stakes for each node
Ensure: Network contains at least 1 node
1: Let N_i be the i^{th} node in the network.
2: Let S_i be the stake of the i node in the network.
3: Let AS be the average stake of the N_i nodes of network.
4: **for** every *node* in the *TallyStake* **do**
5: Let Si be the current stake of the Ni node.
6: let TS be the total stake of the network
7: Total stakes $TS_i = TS_i + S_i$
8: **end for**
9: Calculate the Average stake $AS = TS/N$.
10: Calculate the whether the node is strong or weak.
11: **for** *every node* in the *TallyStake* **do**
12: **if** $S_i \geq AS$ **then**
13: N_i node will add to the strong pool
14: **else**
15: N_i node will add to the weak pool
16: **end if**
17: **end for**
18: Sort the pools based on the stakes S_i

Second Stage Game

The strategy of whether the node will choose to attack or not is illustrated in Algorithm 2. Let $MP_i = mp_1, mp_2, \cdots, mp_n$ where MP denotes the N_i node's mining power. The number of blocks mined by the N_i node is used as the mining power in our approach. We also increased the mining power of the nodes by leveraging their stakes and the following formula is used to determine additional mining power.

$$MP = MP + \frac{S_i}{32} \tag{6}$$

Algorithm 2 Second Stage game

Input: A set of nodes N_i in the network
Output: All the nodes are participating in the consensus
1: let N_i be the node of the Network.
2: Let MP_i be the mining power of the N_i node
3: Add mining power for both pools
4: **for** every node in the Both Pool **do**
5: Calculate mining power
6: $MP_i = MP_i + S_i / 32$
7: Trigger the attacking function
8: $S_i = S_i + stakes$
9: Recalculate the mining power
10: **if** MPi < avgMP and MP_i > 1/4th MP of N nodes **then**
11: $S_i = S_i - (S_i * 25)/100$
12: $MP_i = MP_i + (S_i / 32)$
13: **end if**
14: **end for**

Nodes within the network engage in attacks against each other with the aim of becoming a miner and increasing their mining power through the addition of more stakes. When we compare nodes in strong pools with those in weak pools, the likelihood of becoming a miner is consistently elevated because strong pool nodes have higher stakes and greater mining power compared to weak pool nodes $(StrongPool_{MP} > WeakPool_{MP})$. In weak pools, the risk of being targeted in an attack is notably higher. If an individual node's mining power (MP_i) is below the network's average $(MP_i < avg)$ but surpasses one-fourth of the total mining power in the network $\left(MP_i > \frac{1}{4}\right)$, the node will engage in an attack as a form of retribution. Then, we lower the 25 percent stakes of the node that has been picked for attack which results in losing their mining power. They also lose their possibility of becoming a miner. Following the completion of the attacking strategy, the algorithm will select the top two players based on their mining power. We shall choose based on stakes if both players have zero mining power. Table 3 depicts the matrix of two nodes deciding whether or not to attack. The mining power is represented by the matrix values. Let MP_1 and MP_2 represent each player's mining power.

Table 3. Payoff matrix of second stage game

	Attack	Non-Attack
Attack	$(1 - MP_1),\ (1 - MP_2)$	$1 - MP_1,\ MP_2$
Non-Attack	$MP_1,\ 1 - MP_2$	$1, 1$

- when both players decide to attack, the stakes and mining power are reduced. i.e., $(1 - MP_1), (1 - MP_2)$ where

$$\frac{1}{4} < MP_1 < avgMP \tag{7}$$

$$\frac{1}{4} < MP_2 < avgMP \tag{8}$$

- when one node chooses to attack and another chooses not to, the mining power of the sole participant is diminished. i.e., $1 - MP_1,\ MP_2$ or $MP_1,\ 1 - MP_2$ where

$$MP_1 > avgMP \tag{9}$$

$$\frac{1}{4} < MP_2 < avgMP \tag{10}$$

- when both nodes agree not to attack, no mining power is lost, and the value is reset to 1 where

$$MP_1 > avgMP \tag{11}$$

$$MP_2 > avg\ MP \tag{12}$$

Algorithm 3 Third Stage

Input: The top two nodes of the network
Output: Selected Miner
1: let N_1 and N_2 be the nodes from pool
2: let MP_i be the mining power of N_i node
3: let S and W be the strong pool and weak pool
4: let A and NA be the attack and non-attack strategy
5: **for** i^{th} node in the top two nodes **do**
6: **if** N_i nodes is from S pool and chosen strategy NA **then**
7: $MP_i = MP_i + 1;$
8: **else if** N_i nodes is from S pool and chosen A strategy **then**
9: $MP_i = MP_i - 1;$
10: **else if** N_i nodes is from W pool and chosen NA strategy **then**
11: $MP_i = MP_i - 1;$
12: **else if** N_i nodes is from W pool and choose A strategy **then**
13: $MP_i = MP_i - 2;$
14: **end if**
15: **end for**
16: Choose the miner based on mining power

Third Stage

Algorithm 3 demonstrates the steps involved in selecting a miner after two-stage game. Let N_1 and N_2 be the network's payoff nodes. If both nodes have zero mining power, the miner will choose the one with the highest stake. Otherwise, the node with the highest mining power will become the network's miner.

4. Results and Discussions

Within this section, we initially discuss experimental setup and implementation. Additionally, we discuss the results obtained using the proposed work using various performance parameters.

4.1. Experimental Setup

On a physical computer with an i7-9300H core CPU, which operates at 2.40 GHz, an experimental setup was carried out. This device is running Windows 10. In order to setup the multi-node blockchain network, Oracle VM Virtual Box was employed. The Ubuntu 20.04.4 LTS Operating System configured with 32 GB of running RAM and 360 GB of secondary storage was used for experimentation. For Ethereum blockchain, we utilized Geth 1.10.17. Table 4 shows the detailed configurations of the systems used.

Table 4. Configuration and versions of the systems

Components	Software/Language	Version
OS	Ubuntu	20.04.4 LTS
Processor	Intel	i7-9300H CPU @2.4Hz
Blockchain	Ethereum	4.0
Blockchain Client	Geth	1.10.17

4.2. Implementation in Ethereum

We have utilized the official go-ethereum code base for our implementation and have undertaken substantial improvements with a particular focus on the "clique" module. Our primary objective has been the successful integration of the DDPoS (GT) consensus mechanism into the existing consensus algorithm. Modifications are done in consensus folder of go-ethereum codebase. The customized code, which reflects our modifications is available at the link [22].

4.3. Results Analysis

Within this section, we discuss the results obtained using the proposed work using various performance parameters and scenarios.

Impact of Load

The impact of load, often referred as transaction load, in mining within a blockchain ecosystem is significant and can influence various aspects of the mining process and the overall blockchain network. Fig. 2 depicts the impact of load on the transaction time. These results are based on a 35-node network with variable transactions. When the number of transactions is increased from 100 to 500, the process of electing a miner for the network occurs after each miner has mined their block, causing PoS to take longer than other algorithms. The miner election procedure in DPoS will take place only after all of the delegated nodes have finished their mining. The DPoS takes less time to commit a transaction in the Ethereum blockchain than the PoS. In DDPoS (GT), the process of selecting a miner will be limited to only two top nodes. As a result, it will take less time than the other two algorithms.

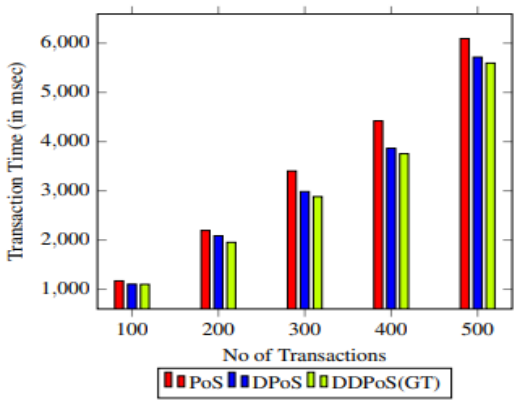

Fig.2. Transaction time v/s number of transactions

Impact of Network Size

The network size in blockchain, particularly in the context of mining, has several significant impacts on the mining process and the overall blockchain network. Fig. 3 depicts the effect of network size on the transaction time. In this scenario, the number of nodes is from varied from 5 to 35 keeping 100 transactions as constant. As shown in Fig. 3, DDPoS (GT) consensus requires shorter transaction time than PoS and DPoS consensus for the same number of transactions, i.e., 100. Furthermore, it is observed that transaction time improvement of DDPoS (GT) is more in this scenario compared to varying transaction scenario discussed in the previous section. This is due to the fact that varying the number of nodes makes mining process more realistic and accurate.

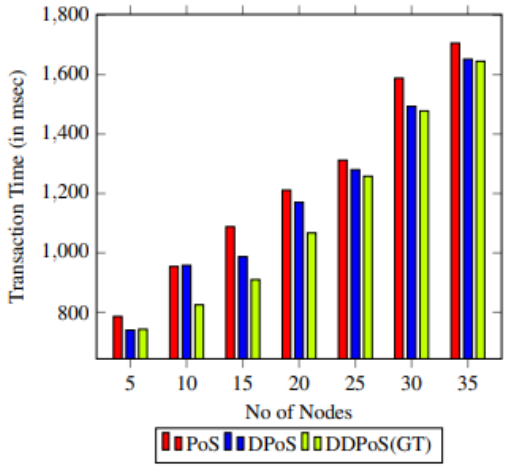

Fig.3. Transaction time v/s number of nodes

Fairness among Nodes

Fig. 4 shows the percentage of the number of blocks mined by each node in the network. It is apparent that the chance of each of the nodes to get the miner status is almost equal, thus the algorithm is fair because miners are prevented from mining the block repeatedly. This is a desirable property of a blockchain network, as it ensures that no single node dominates the network, thereby reducing the risk of centralization and increasing the security of the network. The Fairness Index is calculated using Jain's Fairness Index formula.

$$J(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot x_i^2} \tag{13}$$

Here, *n* is the percentage of blocks generated by each node. The Fairness Index (FI) of the DDPOS(GT) algorithm as derived from the values of Fig. 4 is 0.97 which performed better than the FI of the POS in Ethereum i.e., 0.90. This is due to the fact that in POS algorithm, the opportunity to become a miner is reserved exclusively for the node with the highest stake. Conversely, in DPoS, only delegated nodes have a chance at mining the network. However, when applying game theory principles, the miner's selection depends on both mining power and stakes. With each iteration of the game, a new miner for the network is chosen. As a result, DDPOS (GT) performs better than POS.

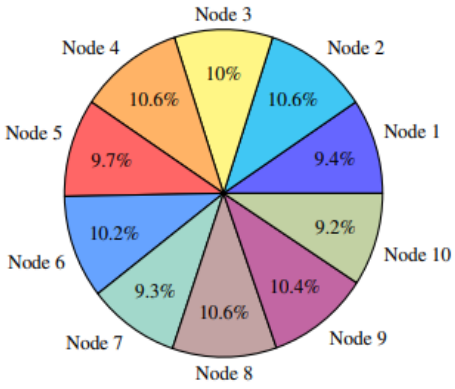

Fig.4. Block percentage of every node in DDPOS(GT)

Malicious Node Behavior

Malicious miners may try to manipulate the mining process to gain an unfair advantage, such as withholding mined blocks to release them strategically. Fig. 5 depicts the results of the downgrading mechanism for a 10-node scenario. In our implementation, we randomly choose some malicious nodes using random function. In DPoS, if a delegated node behaves maliciously, there might not always be a direct punishment mechanism in place. However, in a DDPoS (GT), nodes that engage in malicious behavior face lower stakes as a consequence. It is important to note that this model extends its impact beyond just delegated nodes; it affects all nodes in the network. Therefore, when it comes to deterring and penalizing malicious nodes, DDPoS (GT) is performs better than traditional DPoS, as the consequences of malicious actions extend to all network participants due to the game theory-based incentives.

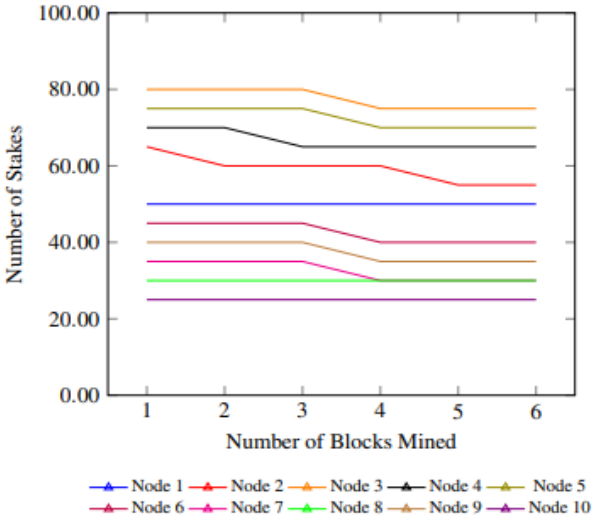

Fig.5. Malicious node behavior and downgrading in DDPOS(GT)

Average Waiting Time

The average waiting time in blockchain refers to the average amount of time it takes for a transaction to be included in a block and confirmed on the blockchain network. This metric is essential for assessing the efficiency and speed of a blockchain network. Fig. 6 illustrates the average waiting time for nodes in a network with 10 nodes and 100 transactions. In DPoS algorithm, a node must wait until all delegated nodes have completed their mining processes before it can participate. This waiting time can be significant leading to delays in transaction processing. In contrast, in a DDPoS (GT), a node has the potential to become a miner for the next block if it possesses more mining power than other nodes. This means that a node has a higher likelihood of quickly becoming a miner, reducing its waiting time significantly. As a result, DDPoS (GT) performs better than DPoS in terms of reducing the waiting time for nodes, which can lead to faster transaction processing and a more efficient network.

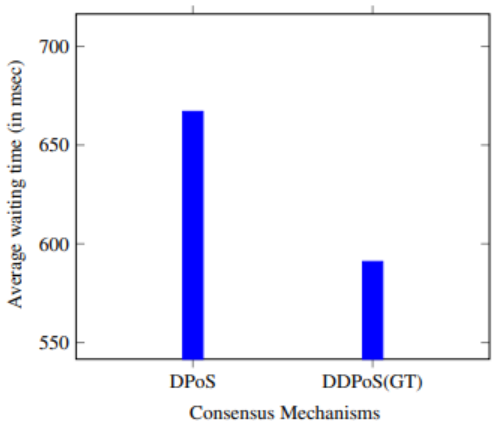

Fig.6. Average waiting time v/s consensus mechanisms

5. Conclusions and Future Work

In this work, we leveraged game theory to devise a new consensus mechanism known as Delegated Proof of Stake with Downgrading to address the limitations of existing consensus protocols such as PoS and DPoS. Our approach involved a three-stage game design. In the first stage, we organized nodes into groups based on their stakes and eligibility for mining. This division set the stage for subsequent decision-making processes. In the second stage, nodes were presented with the option to launch attacks to enhance their chances of becoming network miners. This strategic element aimed to optimize the selection of node miners. Once the second stage was completed, the final stage of the game focused on the selection of a miner. Here, we employed a two-player game to choose the top two nodes, with the selected miner responsible for mining the network and contributing blocks to the blockchain. In the real-time Ethereum network, our findings demonstrate that DDPOS(GT) performs better than other POS and DPOS mechanism in terms of both average transaction time and average block waiting time. Specifically, DDPOS(GT) leads to 10 % reduction in transaction time compared to POS and a 4 % reduction compared to DPOS. Additionally, the average block waiting time experiences a decrease of 11 % when compared to DPOS. These results highlight the efficiency and effectiveness of DDPOS(GT) in optimizing transaction processing and block confirmation times within the Ethereum network.

In our future work, we intend to conduct a comprehensive evaluation of the consensus mechanism by considering additional performance metrics and scaling it to accommodate a significantly larger number of nodes.

References

[1] Guo, Huaqun, and Xingjie Yu. "A survey on blockchain technology and its security." Blockchain: research and applications 3, no. 2 (2022): 100067.

[2] Bamakan, Seyed Mojtaba Hosseini, Amirhossein Motavali, and Alireza Babaei Bondarti. "A survey of blockchain consensus algorithms performance evaluation criteria." Expert Systems with Applications 154 (2020): 113385.

[3] Sethi, Prateek. "Reinforcement Learning assisted Adaptive difficulty of Proof of Work (PoW) in Blockchain-enabled Federated Learning." PhD diss., Virginia Tech, 2023.

[4] Yun, Jusik, Yunyeong Goh, and Jong-Moon Chung. "Analysis of mining performance based on mathmatical approach of PoW." In 2019 International conference on electronics, information, and communication (ICEIC), pp. 1-2. IEEE, 2019.

[5] Feng, Weijia, Zhenfu Cao, Jiachen Shen, and Xiaolei Dong. "RTPoW: A Proof-of-Work Consensus Scheme with Real-Time Difficulty Adjustment Algorithm." In 2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS), pp. 233-240. IEEE, 2021.

[6] Zhang, Shulai, and Xiaoli Ma. "A general difficulty control algorithm for proof-of-work based blockchains." In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3077-3081. IEEE, 2020.

[7] Yu, Bin, Xiaofeng Li, and He Zhao. "PoW-BC: A PoW Consensus Protocol Based on Block Compression." KSII Transactions on Internet & Information Systems 15, no. 4 (2021).

[8] Vasin, Pavel. "Blackcoin's proof-of-stake protocol v2." URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf 71 (2014).

[9] Saleh, Fahad. "Blockchain without waste: Proof-of-stake." The Review of financial studies 34, no. 3 (2021): 1156-1190.

[10] Li, Aiya, Xianhua Wei, and Zhou He. "Robust proof of stake: A new consensus protocol for sustainable blockchain systems." Sustainability 12, no. 7 (2020): 2824.

[11] Yang, Fan, Wei Zhou, QingQing Wu, Rui Long, Neal N. Xiong, and Meiqi Zhou. "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism." IEEE Access 7 (2019): 118541-118555.

[12] Tan, Chao, and Liang Xiong. "DPoSB: Delegated Proof of Stake with node's behavior and Borda Count." In 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), pp. 1429-1434. IEEE, 2020.

[13] Wang, Baocheng, Zetao Li, and Haibin Li. "Hybrid consensus algorithm based on modified proof-of-probability and DPoS." Future Internet 12, no. 8 (2020): 122.

[14] Niya, Sina Rafati, and Burkhard Stiller. "Bazo: A proof-of-stake (pos) based blockchain." IFI-TecReport No. 2019.03, Zürich, Switzerland, Tech. Rep. (2019).

[15] Andreina, Sébastien, Jens-Matthias Bohli, Ghassan O. Karame, Wenting Li, and Giorgia Azzurra Marson. "Pots: A secure proof of tee-stake for permissionless blockchains." IEEE Transactions on Services Computing 15, no. 4 (2020): 2173-2187.

[16] Kumar, Adarsh, and Saurabh Jain. "Proof of game (PoG): A game theory based consensus model." In Sustainable Communication Networks and Application: ICSCN 2019, pp. 755-764. Springer International Publishing, 2020.

[17] Zhang, Bowen, Yucheng Dong, Hengjie Zhang, and Witold Pedrycz. "Consensus mechanism with maximum-return modifications and minimum-cost feedback: A perspective of game theory." European Journal of Operational Research 287, no. 2 (2020): 546-559.

[18] Boreiri, Zahra, and Alireza Norouzi Azad. "A novel consensus protocol in blockchain network based on proof of activity protocol and game theory." In 2022 8th International Conference on Web Research (ICWR), pp. 82-87. IEEE, 2022.

[19] Lin, Zhongjie. "Consensus based on learning game theory with a UAV rendezvous application." Chinese Journal of Aeronautics 28, no. 1 (2015): 191-199.

[20] Kim, Sungwook. "Two-phase cooperative bargaining game approach for shard-based blockchain consensus scheme." IEEE Access 7 (2019): 127772-127780.

[21] Bala, Kirti, and Pankaj Deep Kaur. "A novel game theory based reliable proof-of-stake consensus mechanism for blockchain." Transactions on Emerging Telecommunications Technologies 33, no. 9 (2022): e4525.

[22] Modified source code of Go-Ethereum at URL: https://github.com/0xminer11/go-ethereum/tree/DDPoS(GT).

Authors' Profiles

Naveen Arali has received her BE in Computer Science and Engineering from KLE Technological University, Hubballi, India in 2022. Currently, he is working as Blockchain developer in Chaincode Consulting LLP, Bangalore, India.

Dr. Narayan D. G. is currently working as a Professor in the School of Computer Science and Engineering at KLE Technological University, Hubballi, India. He obtained Ph.D. and M.Tech. from Visvesvaraya Technological University, Belgaum in 2017 and 2004 respectively. He obtained B.E. in Computer Science and Engineering from Karnataka University in 2000. He has 23 years of teaching and research experience. His research interest includes wireless mesh networks, cyber security, cloud computing, software defined networks and blockchain. He is a reviewer for 30 SCIE indexed journals. He has published more than 80 papers in journals and conferences.

Mr. Altaf Husain M. is working as Assistant Professor at School of computer science and engineering, KLE Technological University, Vidyanagar, Hubballi. He has pursued his Master of Technology in Computer Science in 2016 from VTU Belgaum and Bachelor of Engineering from Karnataka university in 2010. He has 12 years of experience. His areas of interest include cloud computing and blockchain technology.

Dr. Prakash S. Hiremath has obtained M.Sc. degree in 1973 and Ph.D. degree in 1978 in Applied Mathematics from Karnataka University, Dharwad. He had been in the Faculty of Mathematics and Computer Science of various institutions in India from 1977 till 2014. Presently, working as Professor, Department of Computer Science (MCA), KLE Technological University, Hubballi, Karnataka, India. His research areas of interest are Optimization Techniques, Image Processing and Pattern Recognition and Computer Networks. He has published more than 250 research papers in peer reviewed International Journals and Proceedings of International Conferences.

How to cite this paper: Naveen Arali, Narayan D. G., Altaf Husain M., P. S. Hiremath, "An Efficient and Secure Blockchain Consensus Algorithm Using Game Theory", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.2, pp.92-102, 2024. DOI:10.5815/ijcnis.2024.02.08