DPoSEB: Delegated Proof of Stake with Exponential Backoff Consensus Algorithm for Ethereum Blockchain

Narayan D. G., Naveen Arali, R. Tejas

Abstract

Blockchain is a technology that is rapidly gaining prominence and finding applications in various sectors such as banking, supply chain, healthcare, and e-governance. The consensus algorithm employed in a blockchain network is crucial as it directly impacts the network's performance and security. Different consensus techniques exist, including Proof of Work (PoW), Proof of Stake (PoS), Robust Proof of Stake (RPoS), and Delegate Proof of Stake (DPoS), each with its own set of advantages and disadvantages. In this work, we propose a new consensus algorithm called Delegated Proof of Stake with Exponential Back-off (DPoSEB). DPoSEB utilizes a stake-based selection of delegates and employs an exponential back-off technique to mitigate collisions among nodes within the network. Each delegate is assigned a random sleep time, and the node with the shortest wake-up time is chosen to mine the block for that particular round. However, collisions among nodes can still occur. To provide a fair chance for each delegate node, collided nodes are assigned an exponential back-off time. We implement our proposed algorithm on an Ethereum-based private blockchain network. To evaluate the effectiveness of our proposed work, we compare it with existing consensus mechanisms such as PoS (version 2) and Delegated RPOS with downgrading (DDRPOS) using different scenarios in terms of transaction latency, waiting time, and fairness as evaluation metrics. The results reveal that DPoSEB performs better than POS and DDRPOS.

Keywords: Blockchain, Consensus, POW, POS, DPOS, DDRPOS, DPoSEB, Ethereum

MSC 2020: 68M12, 68M14, 68W15

ACM CCS 2020: C.2.4

doi:10.56415/csjm.v32.14

1 Introduction

Blockchain technology is based on a growing list of interconnected records known as blocks, which are secured through cryptographic methods. Initially utilized by the cryptocurrency Bitcoin to record transactions, each block in a blockchain contains a cryptographic hash of the previous block, along with a timestamp and transaction data. This design ensures that the stored data remains tamper-resistant. Operating as an open and distributed ledger, blockchain enables efficient, verifiable, and permanent recording of transactions between parties within 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. The decentralized approach employs cryptographic hash functions to establish links between blocks, while a Merkle tree is used to maintain the integrity of all the transactions contained in the block.

Because blockchain functions in a decentralized manner, lacking a centralized authority for reliable governance, anyone interested in participating in a permissionless public network can become a member. With the absence of a trustworthy central authority, individuals can serve as nodes, and all nodes within the network are considered untrustworthy. Hence, a consensus mechanism becomes essential for all nodes to reach an agreement on the network's status. This mechanism establishes a unified perspective on the ledger's global valid state among all network nodes at any given time. The chosen consensus protocol significantly influences blockchain performance, affecting factors such as efficiency, fairness, security, and integrity [1] [2].

Ethereum and Bitcoin are popular blockchain platforms that utilize consensus mechanisms like Proof of Work (PoW) and Proof of Stake (PoS) to reach an agreement on the state of the network. Although many new consensus mechanisms based on PoW and PoS have been introduced, they each come with their own challenges related to scalability and security [3]. PoS brings about certain challenges, particularly concerning security vulnerabilities and the possibility of wealth concentration among stakeholders. Delegated Proof of Stake (DPOS) was introduced to improve network scalability, but it carries risks to fairness and overall network security, particularly if the delegated nodes are compromised. Additionally, as most of the work in the literature focuses on the design and evaluation of consensus algorithms through simulations, there is a need for the design and experimental evaluation of new consensus algorithms in real-time blockchain platforms.

In this work, we propose a new consensus algorithm called Delegated Proof of Stake with Exponential Back-off (DPoSEB) to create a blockchain network that is secure and fairer. In the proposed algorithm, delegate nodes are selected based on the number of stakes and the coinage. Next, each delegate node in the network generates a random wait time after every mining iteration. The wait time represents the amount of time a node must wait before being eligible to propose a new block. The delegated node with the shortest wait time becomes the miner and is eligible to propose a new block. During this process, if the two node's wait time is the same, we refer to this as the collision. To avoid nodes generating the same random wait time next time, we use the exponential backoff algorithm to increase the random wait time. We also identify and penalize malicious nodes by slashing their stake. We implement our proposed algorithm on a real-time Ethereum-based private blockchain network. We evaluated the proposed algorithm by comparing it with existing consensus mechanisms such as PoS (version 2) and DDRPOS, using performance parameters such as transaction latency, waiting time, and fairness as evaluation metrics. The contributions of the work are as follows:

- We analytically derived the average waiting time for a block in the Ethereum blockchain
- We proposed a new consensus named Delegate Proof of Stake with Exponential Back-off (DPoSEB) based on the analytical model
- Evaluated the proposed algorithm with PoS (version 2) and DDR-POS algorithms in terms of different performance metrics using various scenarios in an Ethereum-based blockchain.

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

2 Related Work

The authors in [4] give a brief introduction to blockchain technology and its fundamental components, including consensus algorithms. They highlight two primary types of consensus algorithms: proof-based and vote-based. The authors of [5] 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. They proposed the Proof of Work (PoW) consensus method 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 [6], 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 [7], 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.

In [8], the authors investigate a two-layer neural network algorithm designed to regulate block difficulty. The study considers parameters such as fast updation, low volatility, and hash rate. However, a limitation is noted: the overall accuracy of the neural network falls below 90 %. The authors conduct a Monte-Carlo simulation of the algorithm, utilizing real data from Ethereum. In [9], 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 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 [10], 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 [11], 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 there is a need for a reward mechanism in the design of stake-based consensus algorithms.

In [12], the authors propose the Robust Proof of Stake (RPoS) 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 [13], the authors introduce behavioral credits and establish credit ratings that interact with the currency age in the PoS mechanism. By doing so, they aim to achieve a more equitable and reasonable distribution of revenue among nodes participating in the PoS mechanism.

In [14], the authors focus on improving the efficiency of blockchain and reducing resource consumption. Specifically, they examine the Delegated Proof of Stake (DPoS) consensus mechanism and propose an optimization scheme to address its shortcomings. The proposed scheme, called DPoSB, tackles the issue of block generation failures in DPoS and reduces the likelihood of a malicious node being elected as a witness node. To achieve this, Borda Count is utilized to conduct preference score statistics on candidate nodes. In [15], the authors discuss the block generation and validation processes in the context of two consensus algorithms: the modified Proof-of-Probability (PoP) consensus algorithm and the DPoS consensus algorithm. The modified PoP nodes perform a modulo operation on the nonce value, which is then compared with the expected value provided by the supernode selected by the DPoS nodes.

In [16], the authors acknowledge that Proof of Stake (PoS) is a prominent candidate for resolving the energy demand issue associated with existing blockchain protocols like Bitcoin and Ethereum. The proposed protocol suggests selecting a random node for block mining based on the stake it holds in the blockchain network. This approach reduces energy consumption. However, it also introduces new challenges that were not present in Proof of Work (PoW)-based blockchains. In [17], the authors discuss changes made to the forging algorithm of Waves, a blockchain platform. These changes aim to improve the fairness of block generation and enhance resistance against multi-branching attacks. They describe the current Proof of Stake algorithm, highlighting its limitations. They also address improvements in Nxt's PoS model and adjustments that can be made to enhance it. The authors conduct experiments involving Proof of Stake attacks and propose algorithmic enhancements to mitigate these attacks.

In [18], the authors introduce BAZO, a Proof of Stake (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 [19], 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 [20], the author introduces a consensus technique called Proof of Game (PoG) that can be applied to both single-player and multi-player challenges. The author suggests that a multi-round challenge enables a device with limited resources to achieve high security within a short timeframe. Additionally, they highlight that the presence of selfish miners can lead to a significant decrease in the number of mined blocks as computational difficulty increases, posing challenges to blockchain operations. In [21], the authors present a bi-level optimization model based on the Stackelberg game to capture the interaction between decision-makers and a moderator. They propose a consensus mechanism called Maximum-Return Modifications and Minimal-Cost Feedback (MRMCCM). The MRMCCM approach considers equilibrium strategies, including moderating and compensating tactics that involve the best possible proposed viewpoint and unit consensus cost. The authors address the bi-level optimization model using adaptive differential evolution and conduct extensive experiential experiments to demonstrate the effectiveness of MRMCCM. In [22], the authors introduce the Proof of Activity (PoA) protocol, which incorporates game theory. They devise a unique consensus approach that 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 [23], 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 [24], 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 [25], the authors propose a systematic distributed optimization strategy based on the concept of fictional play. They demonstrate the algorithm's convergence under the context of game theory. The outcome resembles a consensus problem, providing a new perspective on addressing consensus challenges. Numerical instances are used to illustrate the applicability of the proposed strategy.

While most of the works carried out in [3]-[23] focus on implementing and evaluating consensus algorithms through simulations, this work takes a real-time testbed approach. Furthermore, based on the literature review, Table 1 provides a summary of the characteristics of popular consensus algorithms and highlights the research gaps. The table shows that each algorithm has its own advantages and disadvantages. However, existing algorithms require improvements in terms of security and fairness among nodes in achieving consensus. To address the research gaps, we propose a new consensus mechanism called Delegated Proof of Stake with Exponential Back-off (DPoSEB) which enhances fairness among nodes in the network while ensuring network security.

Table 1. Research Gaps Identified

	Consensus Algorithms		
Property	**PoW**	**PoS**	**DPoS**
Blockchain type	Permissionless	Both	Both
Consensus Mechanism	Computational puzzle	Validators with stakes	Delegates with Stake
Efficiency	Low	Medium	High
Security	High	Low	Medium
Fairness	High	Low	Medium
Scalability	Low	Medium	High

3 Proposed Methodology

In this section, we initially discuss the analytical model for the Ethereum blockchain. Next, we discuss the proposed system model based on this study. Furthermore, we also discuss the algorithms used in the implementation.

3.1 Analytical Study of Ethereum blockchain

Our proposed secure and efficient consensus algorithm makes use of the analytical model to estimate the average waiting time for the blocks. The model makes the following assumptions:

1. The time interval Δx is small enough that only one block can be processed during each mining iteration. In the Ethereum blockchain, we set $\Delta x = 1$ millisecond.
2. The arrival of blocks for any miner is governed by a rate parameter μ, while the departure of blocks after confirmation or non-confirmation is governed by a rate parameter λ.

Based on the above assumptions, we can observe four different states for each mining node.

- The probability of the arrival of one block $= \mu.\Delta x$
- The probability of no block arrival $= 1 - \mu.\Delta x$
- The probability of orphan block (one departure) $= \lambda.\Delta x$
- The probability of no orphan block (no departure) $= 1 - \lambda.\Delta x$

Suppose there are n blocks present at time x, and let $Zn(x)$ be the probability that these blocks are part of the blockchain. If the time is increased from x to $x + \Delta x$, there are three possible outcomes:

$$Z_m(x + \Delta x) = \begin{cases} Z_m(x)(1 - \mu.\Delta x)(1 - \lambda.\Delta x) \\ Z_{m+1}(x)\lambda.\Delta x \\ Z_{m-1}(x)\lambda.\Delta x \end{cases} . \tag{1}$$

By reorganizing the conditions presented in Eq. (1), we get:

$$\begin{aligned} Z_m(x + \Delta x) = {} & Z_m(x)(1 - \lambda \cdot \Delta x)(1 - \lambda\Delta x) + \\ & Z_{m-1}(x)\mu\Delta x + Z_{m+1}(x)\lambda\Delta x \ . \end{aligned} \tag{2}$$

Or

$$\frac{Z_0(x+\Delta x)-Z_0(x)}{\Delta x} = -\mu Z_m(x) - \lambda Z_m(x) + \mu Z_{m-1}(x) + \lambda Z_{m+1}(x) \ . \tag{3}$$

As

$$\lim_{\Delta x \to 0} \left\{ \frac{Z_m(x+\Delta x)-Z_m(x)}{\Delta x} \right\} = 0$$

for stable condition, the R.H.S. of Eq. (3) becomes

$$\mathrm{Z}_{m-1}(x)\mu - (\mu+\lambda)Z_m(x) + Z_{m+1}(x)\lambda = 0 \ . \tag{4}$$

The solution to Eq. (4) assumes that there were no requests at time $x+\Delta x$, and this information is derived from the given states.

$$\begin{aligned} Z_0(x+\Delta x) &= Z_0(x)(1-\mu\Delta x) \\ &= Z_1(x)\lambda\Delta x \\ &= Z_0(x)(1-\mu\Delta x) + Z_1(x)(\lambda\Delta x) \\ \left[\frac{(Z_m(x+\Delta x)-Z_m(x))}{\Delta x}\right] &= Z_m(x+\Delta x) \\ = -Z_0(x)\mu + Z(x)\lambda \ . \end{aligned} \tag{5}$$

Therefore, the L.H.S. of Eq. (5) can be expressed as follows:

$$Z_1(x) = \left(\frac{\mu}{\lambda}\right) Z_0(x) \ . \tag{6}$$

By combining equations (4) and (6), we can obtain the following relationship: Mathematical assessment of blocks acceptance in blockchain

$$\left.\begin{array}{c} Z_0(x) = \left(\frac{\mu}{\lambda}\right)^0 Z_0(x) \\ Z_1(x) = \left(\frac{\mu}{\lambda}\right)^1 Z_1(x) \\ Z_2(x) = \left(\frac{\mu}{\lambda}\right)^2 Z_2(x) \\ \vdots \\ Z_m(x) = \left(\frac{\mu}{\lambda}\right)^n Z_m(x) \end{array}\right\} \ . \tag{7}$$

Summation of all the equations:

$$\sum_{i=0}^{n} Z_i(x) = \left\{ \left(\frac{\mu}{\lambda}\right)^0 + \left(\frac{\mu}{\lambda}\right)^1 + \cdots + \left(\frac{\mu}{\lambda}\right)^n \right\} Z_0(x) \ . \tag{8}$$

Applying limits, as $n \to \infty$ and $\frac{\mu}{\lambda} < 1$, L.H.S. becomes 1 and R.H.S. becomes $\left[\frac{1}{\left(1-\frac{\mu}{\lambda}\right)}\right] Z_o(x)$. Thus, equation (8) becomes

$$1 = \left[\frac{1}{\left(1-\frac{\mu}{\lambda}\right)}\right] Z_B(x) \ . \tag{9}$$

By inserting the expression from equation (9) into equation (8), we obtain the following equation:

$$Z_m(x) = \left(\frac{\mu}{\lambda}\right)^n \left(1-\frac{\mu}{\lambda}\right) \ . \tag{10}$$

For a given variable r and sample size n, the average value can be written as

$$\begin{aligned} Q(n) &= \sum_{m\to\infty}^{N} nZ_m(x) \\ &= \sum_{m\to\infty}^{N} \left(\frac{\mu}{\lambda}\right)^n \left(1-\frac{\mu}{\lambda}\right) \\ &= \left(1-\frac{\mu}{\lambda}\right) \sum_{m\to\infty}^{N} \left(\frac{\mu}{\lambda}\right)^n \ . \end{aligned} \tag{11}$$

By elaborating the equation (11), we get

$$\begin{aligned} Q(n) &= \left(1-\frac{\mu}{\lambda}\right) \left\{ \frac{\mu}{\lambda} + 2\left(\frac{\mu}{\lambda}\right)^2 + 3\left(\frac{\mu}{\lambda}\right)^3 + \cdots \right\} \\ &\equiv \left(1-\frac{\mu}{\lambda}\right) \left(\frac{\mu}{\lambda}\right) \left\{ 1 + 2\left(\frac{\mu}{\lambda}\right)^1 + 3\left(\frac{\mu}{\lambda}\right)^2 + \cdots \right\} \ . \end{aligned} \tag{12}$$

Representing Eq.(12) in terms of differentiation,

$$\begin{aligned} Q(n) &= \left(1-\frac{\mu}{\lambda}\right)\left(\frac{\mu}{\lambda}\right)\frac{d}{d\left[\frac{\mu}{\lambda}\right]}\left\{\frac{\mu}{\lambda}+\left(\frac{\mu}{\lambda}\right)^2+\cdots\right\} \\ &\equiv \left(1-\frac{\mu}{\lambda}\right)\left(\frac{\mu}{\lambda}\right)\frac{d}{d\left[\frac{\mu}{\lambda}\right]}\left\{\frac{\frac{\mu}{\lambda}}{1-\frac{\mu}{\lambda}}\right\} \\ &\equiv \left(1-\frac{\mu}{\lambda}\right)\left(\frac{\mu}{\lambda}\right)\left\{\frac{(1-\frac{\mu}{\lambda})+\frac{\mu}{\lambda}}{(1-\frac{\mu}{\lambda})^2}\right\}; \\ Q(n) &= \frac{(\frac{\mu}{\lambda})}{(1-\frac{\mu}{\lambda})}. \end{aligned} \tag{13}$$

Equation 13 provides a way to calculate the average number of blocks that exist in the network at any given time. The mathematical function $W(x)$ represents the predicted time required for a block or blocks to become part of the blockchain network. This function calculates the average waiting time for a block to be processed and added to the blockchain. We use this metric to enhance the performance of the new consensus algorithm.

$$\begin{aligned} W(x) &= \frac{\text{Avg. No. of blocks}}{\text{Block admission rate}} = \frac{Q_n}{\lambda}; \\ W(x) &= \frac{\frac{(\frac{\mu}{\lambda})}{(1-\frac{\mu}{\lambda})}}{\lambda} = \left(\frac{1}{\lambda}\right)\frac{(\frac{\mu}{\lambda})}{(1-\frac{\mu}{\lambda})}; \\ W(x) &= \frac{1}{\lambda}\left(\frac{\mu}{\lambda-\mu}\right). \end{aligned} \tag{14}$$

3.2 System model

The main procedure for implementing any consensus algorithm involves the mining of a block. This entails obtaining a valid hash by utilizing a nonce value along with the hash value of the previous block, subsequently resulting in the creation of a new block. The process of achieving consensus within the DPoSEB blockchain network is depicted in Figure 1. This consensus mechanism involves the careful selection of delegate nodes from the network, aiming to establish unanimity among the participating nodes. Delegate nodes are chosen based on the stakes and coinage. These delegate nodes possess the exclusive authority to both create and mine blocks within the blockchain network. Once the delegate nodes are chosen, the consensus protocol introduces a random sleep interval for each node within the network. The node with the shortest sleep duration awakens, gaining the privilege of initiating a block and receiving the reward. In cases where two nodes are assigned the same time interval, both node's timers expire at the same time and the collision occurs. To address this, the consensus algorithm implements an exponential backoff period for the nodes, ensuring the seamless continuation of the mining process. The proposed algorithm also detects the malicious nodes and slashes their stake.

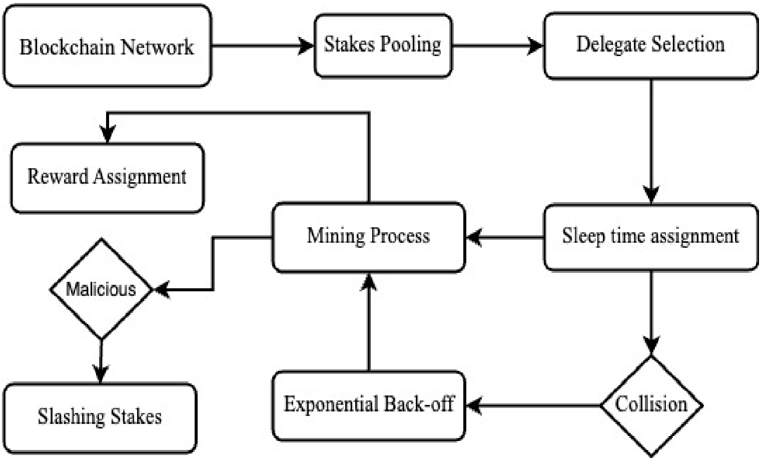

Figure 1. System Model

3.3 Algorithms

In this section, we discuss the algorithms used in the implementation of the proposed work.

Selecting Delegated Nodes:

Algorithm 1 shows the implementation details for the selection of the top 'm' delegate nodes in the blockchain network containing the 'n' number of nodes. The blockchain network is defined as a peer-to-peer network. The nodes in our network are divided into two types: Delegated nodes and Trading nodes. Delegated Nodes will take part in the mining of blocks in the network and the trading nodes are backup for the delegated nodes (if any malicious node is found). For storing the information of nodes present in the network, we have created a TallyStakes structure which contains Stakes, addresses, timestamp, and coinage of every node. DelegateTallyStakes structure contains the Address and stakes of each delegate node. The system will calculate the age of the coin for each node and then select the delegated nodes from the network based on their coinage.

Algorithm 1: Selection of Delegated Nodes

```
Require: Stakes, TimeStamp
Ensure : Network containing at least 1 node
Begin
Let $N_i$ be the $i^{th}$ node in the network.
Let $S_i$ be the stake of the $i$ node in the network.
Let $T_i$ be the timestamp of the stake $S_i$ at the time of creation.
For every node in the $TallyStake$
    Let $t_i$ be the current time to calculate the coinage.
    Age of the node $A_i$ = $T_i$ - $t_i$
    Coinage $C_i$= $A_i$ * $S_i$
    Appending the coinage $C_i$ in the TallStake structure
EndFor
Sorting the TallyStake structure based on the Coinage $C_i$
  appended for each node $i$
For i=0 to m
    $D_i = N_i$
EndFor
End Begin
```

DDRPOS (RR) Consensus Algorithm:

Algorithm 2 illustrates the operational specifics for achieving consensus within a blockchain network utilizing the DDRPOS with Round Robin consensus mechanism. Initially, the delegate nodes are computed based on stage and coinage using Algorithm 1. Then, the algorithm selects a miner from the pool of eligible delegates using a round-robin algorithm. Once a miner is chosen, a node can generate multiple blocks depending on the stakes it possesses. The greater the stakes, the more blocks the miner can generate. By employing the round-robin algorithm, each delegate is provided an opportunity to generate blocks within the network and receive rewards.

Algorithm 2: Reaching a Consensus in Blockchain using DDRPOS (RR)

```
Require: Stakes
Ensure : Network containing at least 1 node
Begin
Let $N_i$ be the $i^{th}$ node in the network
Let $S_i$ be the stake of the $i$ node in the network
Let $m$ be the number of delegated nodes
Select the delegated nodes using Algorithm 1
For each mining iteration
    A miner is selected from the Delegated Node List
     The miner is given $n$ chances to mine $n$ number of blocks
     $n$ is proportional to the stakes held by the miner
     $D_i$ starts the mining
     $D_i$ receives the reward
EndFor
End Begin
```

DPoSEB Consensus Algorithm:

Algorithm 3 shows the implementation details for reaching a consensus in a blockchain network using the DPoS with an exponential backoff consensus mechanism. The algorithm illustrates the consensus mechanism of DPoSEB. Based on the minimum number of stakes and the coinage, the delegate nodes are selected. Initially, each participant in the network generates a random wait time after a fixed time interval. The wait time represents the amount of time a node must wait before being eligible to propose a new block. The participant with the shortest wait time becomes the leader and is eligible to propose a new block. The leader proposes a new block containing transactions and broadcasts it to the network. In this process, if the two node's wait time is the same, we refer to this as the collision. To avoid nodes generating the same random wait time, we use an exponential backoff algorithm to increase the random wait time for collided nodes. We also identify and penalize malicious nodes by slashing their stake.

4 Results and Discussion

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 environmental setup was carried out. This device is running Windows 10. In order to build up the multi-node blockchain network, Oracle VM Virtual Box was employed. The multi-node blockchain was set up using the Ubuntu 20.04.4 LTS Operating System and configured with 32 GB of running RAM and 360 GB of secondary storage. For blockchain, we utilized Geth 1.10.17. Table 2 shows the detailed configurations of the systems used.

4.2 Implementation in Ethereum

We have utilized the official go-ethereum codebase for our implementation and have undertaken substantial improvements with a particular focus on the ”clique” module. Our primary objective has been

Algorithm 3: Reaching a Consensus in Blockchain using DPoSEB

```
Require: Stakes
Ensure : Network containing at least 1 node
Begin
Let $N_i$ be the $i^{th}$ node in the network
Let $S_i$ be the stake of the $i$ node in the network
Let $m$ be the number of delegated nodes
For (each mining iteration)
     Select the Delegated nodes using Algorithm 1.
     For i=0 to m
        Let $D_i$ be the $i^{th}$ Delegate Node
         $D_i$ node is assigned with random sleep time $SL_i$
    EndFor
     While each miner has mined the block
         Check for active miner(s)
              // random wait timer for node expires
          If $D_i$ is the only active Delegate
              $D_i$ is selected as a miner
              $D_i$ is credited with reward
          Endif
          // multiple node's timer expires resulting in collision
          If More than 1 $D$ is active
              For each active Delegate
                  Assign Exponential Backoff time
              EndFor
         EndIf
          If $D_i$ is malicious
          Slash $D_i$ stake by 25%
     EndWhile
EndFor
End Begin
```

the successful integration of the DPoSEB consensus mechanism into the existing consensus algorithm. Modifications are done in consensus

Table 2. Configuration of the Systems Used

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

folder of go-ethereum codebase. The customized code, which reflects our modifications is available at the link [26].

4.3 Result analysis

In this section, we discuss the results of three algorithms using different scenarios and various performance parameters as follows.

Impact of Load

Figure 2 illustrates how the transaction latency is influenced by the system load. This latency is computed by maintaining a constant network size of 10 nodes while adjusting the volume of transactions as depicted in the preceding graph. When altering the transaction volume within the range of 100 to 500, it becomes evident that the time required by DPoSEB is notably shorter when compared to that of both PoS (version 2) and DDRPOS (RR) consensus mechanisms, all within the same network size of 10 nodes. This observation underscores the superior performance of DPoSEB over PoS (version 2) and DDRPOS (RR) in terms of transaction processing speed and latency.

Impact of Network Size

Figure 3 illustrates the correlation between network size and transaction latency. These data points are derived by keeping the quantity

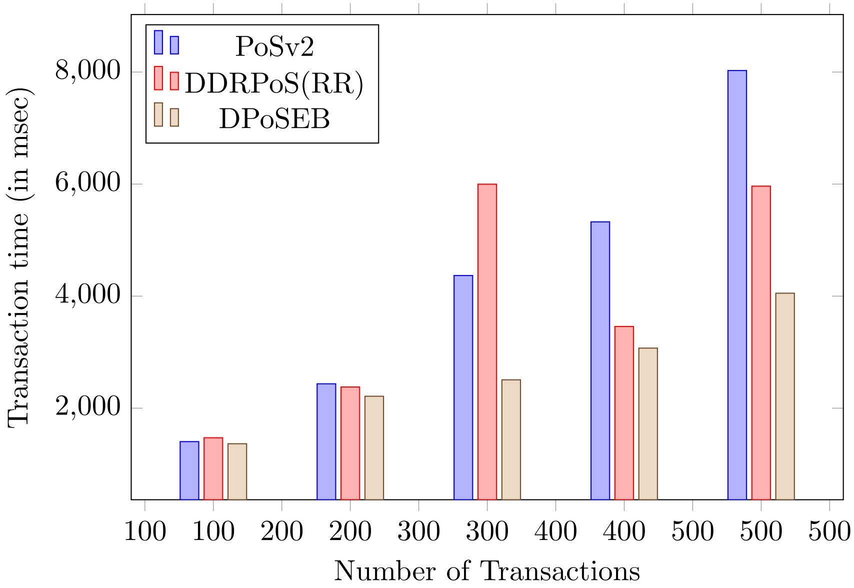

Figure 2. Transaction Latency V/s Number of Transactions

of submitted transactions constant within the network while adjusting the count of nodes present in the network, as demonstrated in the preceding graph. When modifying the number of nodes within the range of 5 to 35, it becomes apparent that DPoSEB exhibits a noticeably shorter time requirement (measured as the average Tx time) in comparison to both PoS (version 2) and DDRPOS (RR) consensus mechanisms. These findings hold true for an identical number of transactions, specifically 100. This observation underscores the superior performance of DPoSEB over PoS (version 2) and DDRPOS (RR) in terms of transaction latency, particularly when considering varying network sizes.

Impact of Network Size on Block Sealing Time

Block sealing time, also known as block time or block interval, refers to the amount of time it takes for a new block to be added to a blockchain. In most blockchain systems, including the popular ones like Bitcoin and

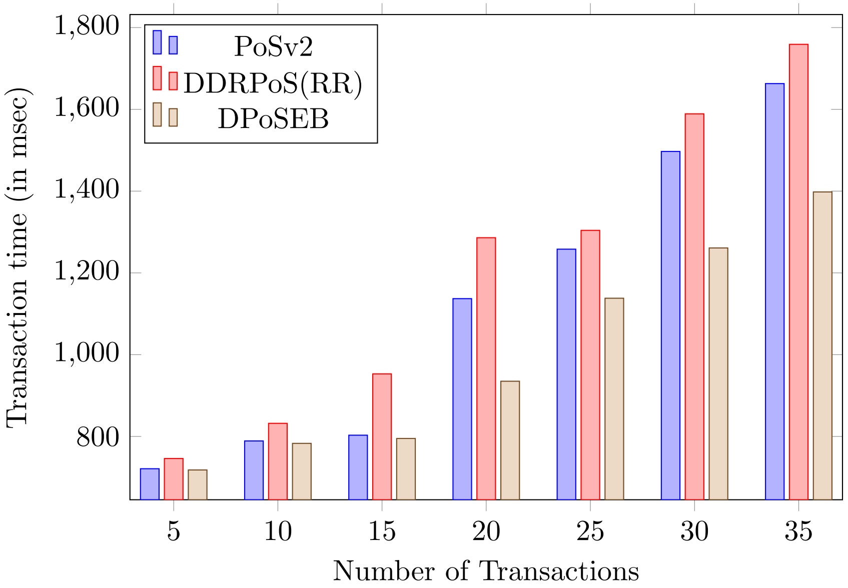

Figure 3. Transaction Latency V/s Number of Nodes

Ethereum, blocks are added to the blockchain at regular intervals. In Figure 4, the influence of network size on block sealing time within the Ethereum network is depicted. The block sealing time pertaining to the transactions submitted within the network is assessed by altering the count of nodes in the network, as illustrated in the graph. When the number of nodes is adjusted within the range of 5 to 35, it becomes evident that the time taken to seal blocks remains nearly identical across all three algorithms, varying only in milliseconds, particularly when dealing with larger network sizes. Conversely, in the context of smaller networks, DPoSEB emerges as notably quicker in sealing blocks compared to both PoS (version 2) and DDRPOS (RR). The outcomes of this analysis highlight that while the sealing times among the three algorithms are relatively consistent with minor differences in milliseconds as the network size increases, DPoSEB distinctly outperforms PoS (version 2) and DDRPOS (RR) in smaller networks by significantly reducing the time required for block sealing.

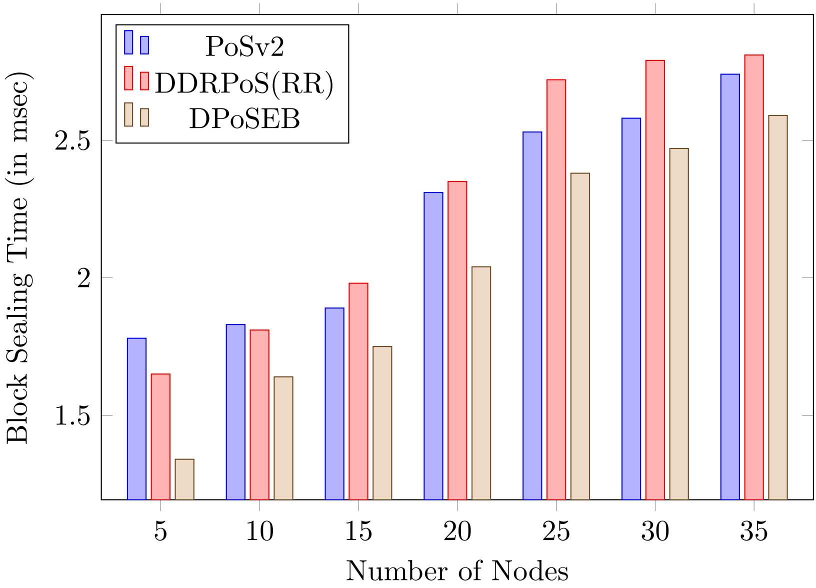

Figure 4. Block Sealing Time V/s Number of Nodes

Fairness among Nodes

We use fairness index to illustrate the fairness of the proposed consensus algorithm. Fairness means every node has an equal opportunity to mine the blocks which is measured by Jain's Fairness Index(J).

$$J(y_1, y_2, \ldots, y_N) = \frac{\left(\sum_{j=1}^{N} y_j\right)^2}{N \cdot \sum_{j=1}^{N} y_j^2} \,. \tag{15}$$

Here, y_j is the percentage of blocks j^{th} node generates. N is the number of nodes mined in the blockchain network. Figure 5 illustrates the fairness index with a varying number of nodes. In the PoS (version 2) consensus mechanism, a miner is selected from eligible miners based on the stake to perform block validation. This approach can result in certain nodes experiencing a starvation scenario, thereby compromising fairness. On the other hand, DDRPOS (RR) ensures fairness by employing a round-robin algorithm, allowing every eligible miner in the network to participate. However, in our implementation, each selected miner mines the number of blocks based on the stake in each round. Thus, it reduces fairness. Furthermore, this approach can lead to extended waiting times for miners due to the potential of multiple blocks being assigned simultaneously to specific miners based on their stakes. In the case of DPoSEB, a unique approach is taken to maintain fairness based on the exponential backup algorithm. All delegate nodes in the network are assigned individual random sleep times. The node with the shortest sleep time gains the mining authority, ensuring that it's able to validate a block. The chosen miner mines a fixed number of blocks, thus, providing a good fairness index. Concurrently, as other nodes awaken from their sleep periods, they are also provided with opportunities to validate blocks. This approach mitigates the possibility of prolonged waiting times and promotes fairness among nodes. Thus, while PoS (version 2) and DDRPOS (RR) exhibit fairness challenges due to high stake node selection and block assignment based on stake, respectively, DPoSEB improves fairness by introducing individual sleep times, guaranteeing equitable opportunities for all nodes to participate in block validation.

Average Waiting Time

The average waiting time in a blockchain network can vary significantly depending on factors such as the network's congestion, and the consensus protocol being used. Figure 6 presents the average waiting time for different consensus algorithms. In DDRPOS (RR), due to the allocation of multiple blocks for mining to certain nodes based on their stakes, an extended waiting time among nodes is observed. This is because some nodes are engaged in handling multiple blocks simultaneously. In contrast, PoS (version 2) encounters waiting time issues as well, owing to the potential starvation of nodes, which leads to a higher waiting time compared to DPoSEB. In DPoSEB, a distinct approach is adopted to address waiting time concerns. Blocks are assigned to nodes randomly, considering their designated sleep times. In situations where a collision occurs, causing multiple nodes to be assigned the same block, an exponential backoff time is introduced. This mech-

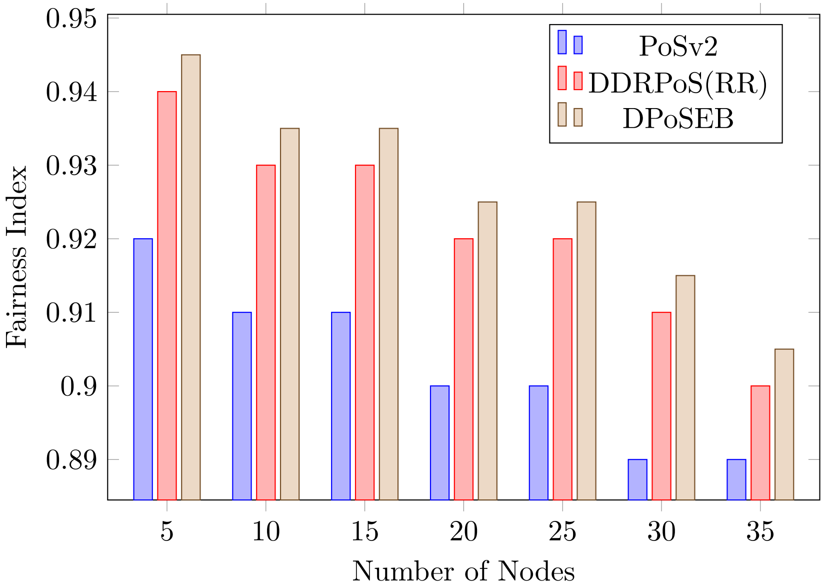

Figure 5. Fairness Index V/s Number of Nodes

anism ensures that miner nodes obtain equitable opportunities within the network and prevents their potential starvation.

5 Conclusion

In this work, we have proposed the consensus mechanism called Delegate Proof of Stake with Exponential Backoff (DPoSEB) to overcome the drawbacks of the existing consensus such as PoW, PoS, and DDR-POS (RR). We use the stakes to select the set of delegates. The selected delegates are responsible for the mining process in the network. We give a random sleep time for each of the delegates, and the node that wakes early is chosen as a miner for that particular round; the same follows for each of the mining processes which improves fairness and decentralizes the network as the right of generating blocks is equally distributed among the delegates. Finally, we add the exponential back-off mechanism to overcome the collision between nodes of the network

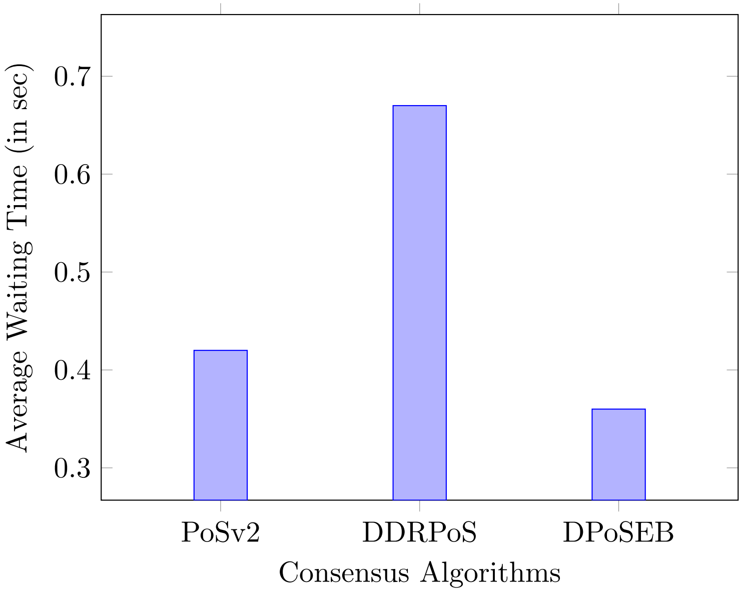

Figure 6. Fairness Index V/s Consensus Algorithms

and give them the chance of mining. From the results, we analyzed that DPoSEB consensus in a permissionless Ethereum network performs better than consensus algorithms PoS (version 2) and DDPoS (RR). The latency of transactions in DPoSEB is less compared to Pos (v2) and DPoS (RR). To handle the collisions, the exponential backoff mechanism performs better in detecting the collisions and assigning a backoff time to the nodes. In future work, we plan to use game theory to improve the fairness in the blockchain network.

References

[1] Huaqun Guo and Xingjie Yu, “A survey on blockchain technology and its security,” *Blockchain: Research and Applications*, vol. 3, no. 2, Article No. 100067, 2022. [Online]. Available: https://doi.org/10.1016/j.bcra.2022.100067.

[2] P. M. Abhishek et al., “Performance evaluation of ethereum and hyperledger fabric blockchain platforms,” in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2022.

[3] Y. Supreet et al., “Performance evaluation of consensus algorithms in private blockchain networks,” in *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*, IEEE, 2020.

[4] Seyed Mojtaba Hosseini Bamakan, Amirhossein Motavali, and Alireza Babaei Bondarti, “A survey of blockchain consensus algorithms performance evaluation criteria,” *Journal of Expert Systems with Applications*, vol. 154, Article No. 113385, 2020.

[5] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, Decentralized business review, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[6] Jusik Yun, 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)*, (Auckland, New Zealand), 2019, pp. 1–2, DOI: 10.23919/ELINFOCOM.2019.8706374.

[7] W. Feng, Z. Cao, J. Shen, and X. 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)*, 2021, pp. 233–240, DOI: 10.1109/ICPADS53394.2021.00035.

[8] S. Zhang and X. 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)*, 2020, pp. 3077–3081, DOI: 10.1109/ICASSP40776.2020.9054286.

[9] B. Yu, X. Li, and H. Zhao, “PoW-BC: A PoW Consensus Protocol Based on Block Compression,” *KSII Transactions on Internet and Information Systems*, vol. 15, no. 4, pp. 1389–1408, 2021. DOI: 10.3837/tiis.2021.04.011.

[10] Pavel Vasin, “Blackcoin’s proof-of-stake protocol v2”, URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper, 2014.

[11] Fahad Saleh, “Blockchain without waste: Proof-of-stake,” *The Review of financial studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[12] Aiya Li, Xianhua Wei, and Zhou He, “Robust proof of stake: A new consensus protocol for sustainable blockchain systems,” *Sustainability*, vol. 12, no. 7, Article ID. 2824, 2020.

[13] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson, “Performance Evaluation of Permissioned Blockchain Platforms,” in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2020, pp. 1–8.

[14] 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*, vol. 7, pp. 118541–118555, 2019.

[15] Chao Tan 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)*, 2020, pp. 1429–1434.

[16] Baocheng Wang, Zetao Li, and Haibin Li, “Hybrid consensus algorithm based on modified proof-of-probability and DPoS,” *Future Internet*, vo. 12, no. 8, Article ID. 122, 2020.

[17] A. Begicheva and A. Kofman, “Fair proof of stake,” Fair Block Delay Distribution, in Proof-of-Stake Project; Waves Platform, 2018, DOI: 10.13140/RG.2.2.11204.37765.

[18] Sina Rafati Niya and Burkhard Stiller, “BAZO: A Proof-of-Stake (PoS) based Blockchain,” IFI-TecReport No. 2019.03, Zürich, Switzerland, Tech. Rep., 2019.

[19] Sébastien Andreina, Jens-Matthias Bohli, G. Karame, Wenting Li, and Giorgia Azzurra Marson, “PoTS: A Secure Proof of TEE-Stake for Permissionless Blockchains,” *IEEE Transactions on Services Computing*, vol.15, no. 4, pp. 2173–2187, 2020.

[20] Adarsh Kumar and Saurabh Jain, “Proof of game (PoG): A game theory based consensus model,” in *International Conference on*

Sustainable Communication Networks and Application, Springer, Cham, 2019, pp. 755–764.

[21] Bowen Zhang, 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*, vol. 287, no. 2, pp. 546–559, 2020.

[22] Zahra Boreiri 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)*, 2022, pp. 82–87. IEEE.

[23] Zhongjie Lin, "Consensus based on learning game theory with a UAV rendezvous application," *Chinese Journal of Aeronautics*, vol. 28, no. 1, pp. 191–199, 2015.

[24] Sungwook Kim, "Two-phase cooperative bargaining game approach for shard-based blockchain consensus scheme," *IEEE*, vol. 7, pp. 127772–127780, 2019.

[25] Zhongjie Lin and Hugh HT Liu, "Consensus based on learning game theory," in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014, pp. 1856–1861. IEEE.

[26] Modified source code of Go-Ethereum at URL: https://github.com/0xminer11/go-ethereum/tree/DPoSEB.

Narayan D. G.[1], Naveen Arali[2],
R. Tejas[3]

Received September 01, 2023
Revised April 30, 2024
Accepted May 27, 2024

[1,2,3] Computer Science and Engineering,
KLE Technological University, Vidyanagar, Hubli, 580031, Karnataka, India.

[1] Narayan D. G.
ORCID: `https://orcid.org/0000-0002-2843-8931`
E–mail: `narayan_dg@kletech.ac.in`

[2] Naveen Arali
E–mail: `naveenarali01@gmail.com`

[3] Tejas R.
E–mail: `tejasrattihalli@gmail.com`