DOI: https://doi.org/10.15276/aait.05.2022.13
UDC 004.9

An incentive system for decentralized DAG-based platforms

Igor Y. Mazurok[1]
ORCID: https://orcid.org/0000-0002-6658-5262; mazurok@onu.edu.ua. Scopus Author ID: 57210121184
Yevhen Y. Leonchyk[1]
ORCID: https://orcid.org/0000-0003-1494-0741; leonchyk@onu.edu.ua. Scopus Author ID: 57192064365
Sergii S. Grybniak[2]
ORCID: https://orcid.org/0000-0001-6817-8057; s.s.grybniak@op.edu.ua
Oleksandr S. Nashyvan[2]
ORCID: https://orcid.org/0000-0001-8281-4849; o.nashyvan@op.edu.ua
Ruslan O. Masalskyi[1]
ORCID: https://orcid.org/0000-0002-8573-9802, masalskyi@stud.onu.edu.ua
[1] Odessa I. I. Mechnikov National University. 2, Dvoryanskaya Str. Odessa, 65082, Ukraine
[2] Odessa Polytechnic National University. 1, Shevchenko Ave. Odessa, 65044, Ukraine

ABSTRACT

Decentralized public platforms are becoming increasingly popular due to a growing number of applications for various areas of business, finance, and social life. Authorless nodes can easily join such networks without any confirmation, making a transparent system of rewards and punishments crucial for the self-sustainability of public platforms. To achieve this, a system for incentivizing and punishing Workers' behavior should be tightly harmonized with the corresponding consensus protocol, taking into account all of its features, and facilitating a favorable and supportive environment with equal rights for all participants. The main purpose of rewards is to incentivize Workers to follow the protocol properly, and to penalize them for any type of misbehavior. The issues of block rewarding and punishing in decentralized networks have been well studied, but the DAG referential structure of the distributed ledger forces us to design methods that are more relevant. Since referential structures cannot be reliably validated due to the fact that they are built on the basis of the instantaneous visibility of blocks by a certain node, we propose to set rewards for blocks in the DAG network based on the degree of confidence of topological structures. In doing so, all honest nodes make common decisions based only on information recorded into the ledger, without overloading the network with additional interactions, since such data are always identical and available. The main goal of this work is to design a fair distribution of rewards among honest Workers and establish values for penalties for faulty ones, to ensure the general economic equilibrium of the Waterfall platform. The proposed approach has a flexible and transparent architecture that allows for its use for a wide range of PoS-based consensus protocols. The core principles are that Workers' rewards depend on the importance of the conducted work for block producing and achieving consensus and their penalties must not be less than the potential profit from possible attacks. The incentivizing system can facilitate protection from various kinds of attacks, namely, so-called Nothing-at-stake, Rich-get-richer, Sybil, and Splitting attacks, and from some specific threats related to a DAG structure.

Keywords: Tokenomics; incentivizing; blockchain; directed acyclic graph; consensus protocol

For citation: **Mazurok I., Leonchyk Y., Grybniak S., Nashyvan O., Masalskyi R. "An incentive system for decentralized DAG-based platforms."** ***Applied Aspects of Information Technology*****. 2022; Vol. 5 No. 3: 196–207. DOI: https://doi.org/10.15276/aait.05.2022.13**

INTRODUCTION

This work deals with the incentivization of nodes of the Waterfall platform to honestly perform their duties for achieving a sustainable, secure, and high-performing network, by driving behaviors of all participants with economic leverages. However, it can be considered as a standalone work that presents an incentive system that can be implemented, in part or in whole, to other Proof-of-Stake (PoS) [1] consensus protocols of decentralized networks.

The issues of creating a fair distribution of rewards among platform Workers and setting values of penalties are addressed in detail.

The incentive mechanism is the backbone of any tokenomics system (*tokenomics* is a term that captures a token's economics). It should facilitate nodes' positive actions such as processing transactions, validating blocks, and finalizing the ledger. We should note that users can join or leave public networks at their own discretion. Obviously, if rewards do not cover Workers' expenditures or are distributed unfairly, honest participants have no incentive to participate in such a network. A good tokenomics practice includes building a community around a project, to discuss emerging challenges for improving the economic environment.

In developing a reward system, special attention should be paid to such questions as:

- What types of work should and can be rewarded to optimize network performance?
- Should rewards depend on the quality of work done?
- Do reward amounts reflect the efforts made and/or the degree of responsibility in the overall work?
- Are the possibilities for honest and reliable nodes to get benefits the same across the board?
- Is the faithful following of a consensus protocol by nodes more profitable to them compared to other behavior strategies?

Generally, some network Workers may not be entirely reliable. For example, they can be off-line (disconnected) for long periods of time or delay connecting with others, reducing the overall performance of the network. Moreover, some Workers may maliciously threaten network security. Hence, both rewards for productive Workers and penalties for faulty Workers play key roles in the operation of public peer-to-peer systems. This is especially important for PoS-based networks like Waterfall, since their entire security relies on a staking mechanism.

In developing a penalty system, special attention should be paid to such questions as:

- Can some participants gain an unfair advantage over others?
- How can we eliminate potential vulnerabilities?
- Which attacks should and can be penalized?
- Do the values of penalties correspond to the seriousness of attacks?
- How can penalties be used to mitigate various attacks on the network?

All vulnerabilities of decentralized public networks should be examined to promote appropriate protection of the consensus protocol and communication between nodes, improving the robustness and trust of the platform as a whole.

A “negative” reputation system does not make sense for such networks due to “zeroing” – a misbehaving Worker can create a new account from scratch and transfer its stake to the new one. Therefore, the system of penalties and bans should incentivize Workers to be reliable and honest, simultaneously preventing a number of attacks that are resistant to cryptographic methods. At the same time, it needs to be appropriately adjusted to provide punishments duly, without generating excessive load on the network and lowering its performance.

We should note that a “positive” reputation system merits attention: A Worker with a good reputation could gain additional advantages and benefits in the future. However, experiments conducted in the framework of the Theory of Loss Aversion give reason to assume that penalties may in some instances turn out to be more effective than rewards in motivating people to behave in a certain way [2].

RELATED WORKS AND PROBLEM STATEMENT

Public decentralized networks cannot function successfully without researching crypto-economics. In this way, all of them pay attention to incentives and punishments of participants to a greater or lesser extent. Nearly every detailed technical document on the implementation of blockchain technology, especially based on a PoS-consensus like Ethereum 2.0 [3], [4], Polkadot [5], Cosmos [6], IOTA [7], etc, contains a chapter describing how well-behaved nodes are rewarded and misbehaving nodes are penalized with a unique mechanism. The differences between methods are both in the amounts of rewards and penalties and, more to the point, in which actions are rewarded and penalized.

In addition, the issue of incentivizing blockchain Validators is actively discussed by game theory researchers (e.g. [8], [9]). Some methods propose frameworks that could be applied to many PoW and PoS blockchains ([10], [11]) while some methods are tightly integrated into certain types of consensus ([12], [13]). However, both approaches use the fundamental characteristics of blockchain technology and the core principles of game theory to direct participants towards responsible behavior, in accordance with the functional goals of the network.

Thus, the applied problem of tokenomics of a public decentralized system can be formulated in terms of the cryptoeconomics of internal tokens of a particular platform. It consists in building an agreed set of economic rules for manipulating the internal token (or several tokens) of the platform, their emission, burning, taxes, commissions, fines and rewards. The set of rules should cover both the macroeconomics of the platform and the microeconomics of individual nodes or decentralized applications, and ensure their consistency to support platform viability, efficiency, and expansion.

THE PURPOSE AND OBJECTIVES OF THE STUDY

The purpose of the work is to create effective tokenomics for the decentralized platform Waterfall. The key functional characteristics of the platform, which allow solving the tasks assigned to it, are maximum decentralization, viability, stability, and dynamism. Under these conditions, tokenomics should provide a high energy potential and economic attractiveness. To achieve this goal, we need to use such cryptoeconomic mechanisms that would allow the internal token of the system to become the driving force of interactions between tens of thousands of Workers and millions of users (crypto wallets). At the same time, tokenomics should make the destructive behavior of participants economically inexpedient. In this way, we faced the following design and analytical objectives.

- Designs a system for accounting for the useful work performed by Workers and establish a fair distribution of remuneration among conscientious Workers.
- Design a direct system for detecting malicious activities and set fines for perpetrators to ensure the overall economic balance of the platform.
- Find a way to indirectly detect violations and develop probabilistic algorithms for fines and rewards.
- Minimize user fees for transactions.
- Ensure system scalability.

Additionally, it should be noted that the remuneration of workers should functionally depend on the assessment of the importance of the work done for the production of blocks and reaching consensus. At the same time, the penalties should not be less than the potential profit from possible attacks and functionally take into account the assessment of potential damage. The motivation system can help protect against various types of attacks, namely the so-called Nothing-at-stake, Rich-get-richer, Sybil, and Splitting attacks, as well as against some specific threats associated with the DAG structure.

PLATFORM OVERVIEW

Waterfall [14] is a highly-scalable EVM-based smart contract platform for developing various decentralized applications (Dapps). Testnet is currently running on 64 t3.small instances (2 cores, 2Gb RAM) of Amazon. Scalability measurements were made: version 2 showed an average speed of 2,234 tps and version 3 – 3,600 tps. The distributed protocol relies on the Directed Acyclic Graph (DAG) [15], [16] with rapid finality Proof-of-Stake (PoS) consensus. The launch of the public mainnet is scheduled for this autumn.

The platform is composed of two interacting networks – the Coordinating network (blockchain-based) and the Sharding network (blockDAG-based). The BlockDAG part achieves high transaction throughput via parallelized block production, due to the DAG structure. The Blockchain part fixes, linearizes and finalizes the chains of produced transaction blocks. The nodes of the Coordination network will be called Coordinators and the nodes of the blockDAG network will be called Validators. Each Worker consists of two parts, a Coordinator and a Validator, presenting it in corresponding networks.

The timeline is divided into slots, epochs, and eras. Coordinators maintain the register of Validators, and they assign block producers, committee members, and leaders in each slot at the beginning of an epoch.

In addition, the Coordinating network contains information about the approved blocks created on the Sharding networks. Each Validator accompanies its created block with links to all known tip-blocks of the DAG. At the same time, the linearization (ordering) and finalization of the distributed ledger are performed in the Coordinating network, increasing overall security and synchronization.

REWARDS

In Waterfall, each Validator is entitled to create blocks in certain slots of the Shard Network, in accordance with assignments received from the Coordinating Network. The Validator forms a block with pending transactions and distributes it among other Validators that include this same block in the DAG ledger (Fig. 1). If the block is a spine in its slot, Validators send its hash to the Coordinating Network to be finalized. Otherwise, the block waits until another spine block is created in a future slot and links to it to be finalized. It should be noted that there is only one spine block per slot, and each of them must gain a few confirmations in the Coordinating Network to be finally accepted.

Coordinating Network. Block creation is incentivized with minted rewards for each block of the Coordinating Network. According to the rules of the consensus protocol, a few committees (C) participate in every block formation, and each of them has N members chosen from among Coordinators. For the purposes of this paper, it is enough to know that block formation is performed in three stages:

1. Committee members vote on a list of visible unfinalized blocks of the Shard Network to be approved and finalized.

2. An aggregator collects signatures from members of its committee and sends a batch to the current slot leader.

3. The slot leader creates a block in the Coordinating Network based on all collected data.

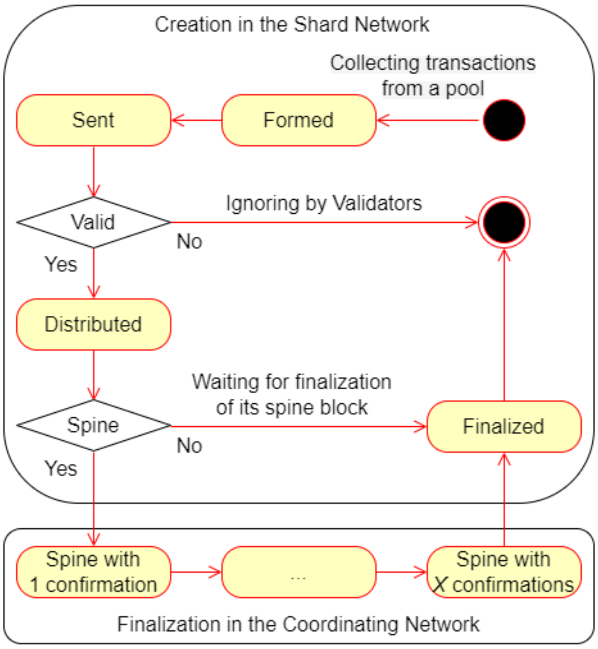

Creation in the Shard Network
Collecting transactions from a pool
Sent
Formed
Valid
Ignoring by Validators
No
Yes
Distributed
Waiting for finalization of its spine block
Spine
No
Finalized
Yes
Spine with 1 confirmation
...
Spine with X confirmations
Finalization in the Coordinating Network

Fig. 1. **Statechart diagram of a transaction block**
Source: **compiled by the authors**

All Coordinators have the same initial stakes as a locked amount of coins, and rewards received are not added to them. However, these stakes may be reduced with penalties over time. A Coordinator may be entitled to participate in committees until its stake is less than 50% of the initial value. These rights are revised for all Workers in every Era. An aggregator is chosen from among ordinary committee members.

Further, we consider that each of the three stages mentioned above is equally important to successfully achieve consensus, and the block reward W is split into three equal parts. Hence, the overall work at each stage will be rewarded by $W/3$.

1. There are $C \cdot N$ committee members per block. Hence, each of them receives

$$v = \frac{W}{3 \cdot C \cdot N}, \qquad (1)$$

in case its vote message is included in a block of the Coordinating Network. It should be noted that the value of v will be further used to define penalties.

2. Each of C aggregators can get

$$\frac{W}{3 \cdot C \cdot N} + \frac{W}{3 \cdot C} \cdot \gamma_1, \qquad (2)$$

where $\gamma_1 \in (2/3, 1]$ is a ratio of included committee members' signatures to the committee size N. Therefore, aggregators are incentivized to collect as many signatures as possible. However, according to the consensus protocol, an aggregator can present a message only if it is signed by more than $2/3$ of committee members. The first component of the sum (2) is received by the aggregator for work as an ordinary committee member.

3. Finally, a slot leader is rewarded by

$$\frac{W}{3 \cdot C \cdot N} + \frac{W}{3} \cdot \gamma_2, \qquad (3)$$

where $\gamma_2 \in (0,1]$ is a ratio of included aggregators' messages. The leader gets the first component of (3) for work as a committee member.

Obviously, if $\gamma_1 = 1$ for all committees in (2) and $\gamma_2 = 1$ in (3), the block reward W is fully distributed among all Workers that participated in the block formation.

The possibilities per slot to be entitled as an ordinary committee member but not an aggregator or leader, an aggregator, and the leader, are equal to $\frac{C \cdot (N-1) - 1}{M}$, $\frac{C}{M}$ and $\frac{1}{M}$ respectively, where M is the total number of Coordinators. Figure 2 depicts the proportion of mathematical expectations of the Coordinator's reward, with $C = 4$ and $N = 64$. In other words, this is the distribution of the Coordinator's reward per Era, based on different types of work.

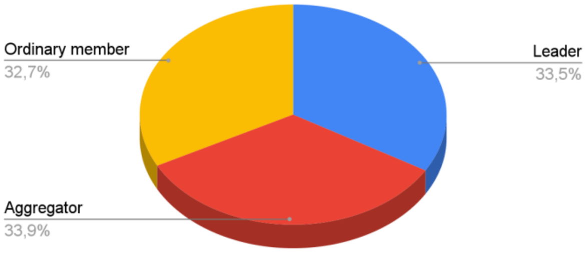


Fig. 2. **The proportion of rewards per Era by work type in the Coordinating Network**
Source: **compiled by the authors**

DAG-based Shard Network. The base transaction fee f for a block is split into two portions with a burning multiplier $l \in [l_0; 1]$:

$$f = l \cdot f + (1 - l) \cdot f, \qquad (4)$$

and the first component is burned but the second is left to a Validator. The parameter $l_0 \geq 0$ represents the minimum portion of the transaction fees that is burned. Therefore, a Validator's reward consists of all transaction tips and a portion of transaction fees, with $l < 1$ included into a produced block. The value of l in (4) can be defined on the basis of the so-called “quality” of the DAG-block.

The main purpose of rewards is to incentivize Workers to follow protocol conscientiously, and to penalize them for cheating attempts or any type of misbehavior. The issue of block rewarding has been well studied, but the DAG structure forces us to design a new mechanism of block rewarding.

A typical task for a DAG network is to maintain a valid referential structure. Having valid references helps to maintain the integrity and security of information in the Shard. However, not all intentional or accidental deviations from the protocol are easy to detect and confirm with a consensus.

We propose a system of rewards based on the behavioral model of honest Validators that is fixed in DAG topology. In doing so, we examined the referential structure of blocks created by honest Workers and built a k-dimensional histogram (where k is the maximum available depth of references) to describe the typical behavior of honest block creators [17].

As a result of modeling, a set of vectors was obtained:

$$B = \{\bar{b} = (b_1, b_2, \dots, b_k)\} \subseteq \aleph^k,$$

where b_i – the number of references with depth i, each of which corresponds to a block created in the Shard Network.

Further, the following histogram g was generated:

$$g(\bar{b}): B \to (0;1], \qquad \textstyle\sum_{\forall \bar{b} \in B} g(\bar{b}) = 1, \qquad (5)$$

that for each vector $\bar{b} \in B$ specifies the relative frequency of its occurrence in the DAG. When constructing this function, we consider that it should not be beneficial for a node to conceal references to tip-blocks known to it. In order to not depend on the degree of detail of the histogram in (5), the function $g(\bar{b})$ is normalized:

$$\hat{g}(\bar{b}) = \frac{g(\bar{b})}{g_{max}}, if\ \bar{b} \in B, else\ \hat{g}(\bar{b}) = 0, \qquad (6)$$

where $g_{max} = \underset{\forall \bar{b} \in B}{max}\, g(\bar{b})$. Then for each produced block $\bar{b}$ we can define the confidence function with the normalized $\hat{g}$ from (6):

$$p(\bar{b}) = \max_{\forall \bar{x} \le \bar{b}} \hat{g}(\bar{x}), \qquad (7)$$

where $\bar{x} \le \bar{b} \Leftrightarrow \forall\, i: 1 < i \le k, x_i \le b_i$. The Validator's reward per block is determined in proportion to the degree of confidence (7), and the burned amount (which can also be considered as a penalty) is inversely proportional to this value. Therefore, such a portion of transaction fees is burned: $l = l_0 + (1 - l_0)\left(1 - p(\bar{b})\right)$ for each block, depending on its referential structure $\bar{b}$.

DAG STRUCTURE MODELLING

Next, we will study the topology of the DAG that is formed during the operation of the protocol. For this purpose, we have developed an appropriate simulation model. The main task of the model was to simulate the work of creating link blocks in the network of validators in one shard of the Waterfall system. Such a model will allow us to investigate the statistical characteristics of the obtained graph (Fig. 3).

According to the current protocol, the simulated system is characterized by the following parameters:

- Slot time is the duration of the time segments into which the network operation time line is divided. During this time, each validator as-signed to this slot can create one block;
- Spray width – the number of blocks spawned in each slot;
- Spray depth – restriction on the depth of links in slots. The depth of generated links to blocks from previous slots does not exceed the value of this parameter.

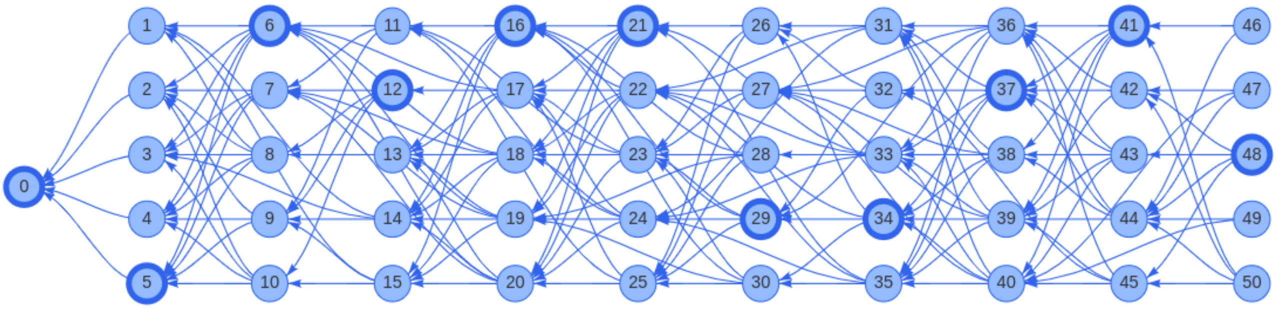


Fig. 3. **DAG structure created with modeling. Vertices in bold are spine blocks.**

Source: **compiled by the authors**

The created parameterized simulation model can be customized by tuning the following parameters:

- Number of slots – Number of slots – the number of time slots in which each Validator has the right to create one block;
- Fault rate – Fault rate – the estimated probability that the node will be faulty (faulty nodes do not create blocks);
- Distribution of time shifts when nodes start to form blocks (relative to the beginning of the slot);
- Distribution of the values of the time spent on the formation of blocks.

Distribution of time required to propagate a block across the network:

- uniform distribution [18] with parameters max, max;
- lognormal distribution [19] with parameters mu, std;
- non-negative normal distribution [20] with parameters mu, std.

The non-negative normal distribution is modeled as repeatedly sampling a value from the normal distribution until it becomes non-negative.

At the first stage of the modeling, a list of tasks is formed. To create a task during each slot, each of the nodes designated for this (according to the width of the spray) creates its own block. The creation of a block starts at a moment in time $slotTime_i + shift_i + timeForCreation_i$. It takes some random time to create a block (its distribution characteristics are set in the system parameters).

After that, the created block is distributed over the network between all honest Validators in the time before $slotTime_i + shift_i + timeForCreation_i + timeForSpreading_i$, where value of parameter $shift_i$ taken from *block start time distribution*, $timeForCreation_i$ taken from *block creation time* distribution, $timeForSpreading_i$ taken from *block spreading time distribution*.

There are two types of tasks. The first type is the task of creating a block starting from the moment of time t_i . The second type consists in spreading the block over the network starting from t_i.

After sorting the task list by start time, we get the sequence of tasks to be processed. Each Validator, the creator of blocks, has a list of blocks still untouched by links (initially it contains only the genesis block). New blocks are referencing to all blocks with a depth no greater than $sprayDepth$ untouched by other known Validators. Distribution updates lists of blocks for each Validator.

CONFIDENCE FUNCTION

The confidence function gives a numerical value of the degree of our confidence that the blocks were created honestly, that is, in accordance with the protocol. On the basis of the results obtained during the simulation, algorithms for calculating this function were investigated.

Let us say that the vector $\bar{b}$ is proper, or $\bar{b} \in K \subseteq \aleph^k$ (where K – set of proper vectors), if it was obtained during the operation of the model. And vice versa vector $\bar{b}$ is not proper ($\bar{b} \notin K$), if it was not obtained during the operation of the model. Then $\forall \bar{b} \in \aleph^k, \hat{g}(\bar{b}) > 0 \Leftrightarrow \bar{b} \in K$.

The following parameters of the model were fixed for the study:

- number of slots – 1000;
- fault rate – 0;
- slot time – 5 sec;
- spray width – 25;
- spray depth – 5;
- block start time distribution as nonnegative normal distribution with parameters: $mu = 1.5$, $std = 0.1, std = 0.1$;
- block creation time distribution as nonnegative normal distribution with parameters: $std = 0.1, std = 0.1$;
- block spreading time distribution as uniform distributions with parameters: $\min = 4.5$, $max = 12$.

As a result of the modeling, 631 proper vectors were obtained. Taking into account the specificity of the input data space of the model, 16945 not proper vectors were obtained.

Since $p(\bar{x}) = max_{\bar{b} \in K \wedge \bar{b} \le \bar{x}} \hat{g}(\bar{b}), \bar{b}, \bar{x} \in \aleph^k$, the problem can be considered as finding the maximum on the prefix tensor constructed at the points $O \in \aleph^k$ (origin of coordinates) and $\bar{x}$. Since there is no need to maintain the update of the function $\hat{g}(\bar{x})$ it is an offline task of finding the maximum on the prefix. Taking this into account, some solutions can be considered.

A naive decision. We will store the entire set K. Upon request, we will bypass the set K and honestly count the maximum.

Solution based on prefix array. Let there be an array $P[M][M]...[M]$, where M – is a constant that limits the space of interest to us. We use the following rules for forming a prefix array:

- $P[i][j]...[q] = 0, i \cdot j \cdot ... \cdot q = 0$;

- $P[i][j]\ldots[q] = \max\{P[i-1][j]\ldots[q], P[i][j-1]\ldots[q], \ldots, P[i][j]\ldots[q-1], \hat{g}((i,j,\ldots q))\}\ 1 \leq i,j,\ldots,q < M.$

Then the request $p(\underline{x})$ is already counted, but can be found in $P[x_1][x_2]\ldots[x_k]$. To implement the prefix array regardless of the dimension of the problem, we performed its linearization and built the corresponding bijective by mapping the index vectors into a scalar index. This approach allows you to avoid solving a multidimensional problem.

Decision based on the octant tree. This solution is based on the principles of the octant tree [21]. Namely, a tree-like data structure is built, in the nodes of which the maximum for the corresponding subtensor will lie.

Solutions based on neural networks [22] that are widely applied in various areas nowadays [23], [24]. Having 17576 pairs of vectors and the corresponding values of the confidence function, which were obtained during the operation of the model, it is possible to construct an approximation of the function $p(\bar{x})$. This task is handled by supervised learning algorithms [25], [26], [27]. The best metrics were obtained using a neural network. A neural network with 1905 weights was studied.

Comparative analysis of confidence function search algorithms. Based on the data presented in the table (Table 1), the following conclusions can be drawn:

- the tree-based solution is unacceptable under any circumstances, as the tree loses to other approaches in terms of memory usage and query execution speed, and does not provide any clear advantages;
- if memory usage is very critical, then given the relatively small number of elements in the set K, you can use a naive solution;
- if the amount of memory used in a solution based on a prefix array is acceptable, then this approach will be the best, because it processes requests faster than all others;
- if there is a need to process many requests at once, while using less memory than a solution based on a prefix array, then a neural network would be a good option.

ATTACKS

In this chapter, the main penalized types of Workers' misbehavior are considered in detail. The penalties are charged automatically on the basis of information recorded in the Coordinating ledger. A core principle is that the penalty must not be less than the potential profit from attacks.

Attacks in the Coordinating Network

In a slot, some members' votes, aggregated messages, or even the block itself can be absent. Obviously, Coordinators missing this slot do not get rewards, but penalties significantly increase the tolerance level for the total number of fault participants, since they are eventually eliminated [6].

Table 1. **Comparison algorithms for confidence function**

	A naive decision	Solution based on prefix array	Decision based on the octant tree	A solution based on a neural network
Memory usage is asymptotic	$O(\lvert K\rvert)$	$O(M^k)$	$O(\lvert K\rvert)$	$O(\lvert K\rvert)$
Memory usage is actual	4.4 kB	70.3 kB	315 kB	7.62 kB
The speed of processing requests is asymptotic	$O(\lvert K\rvert)$	$O(1)$	$O(log\lvert K\rvert)$	$O(\lvert K\rvert)$
The speed of processing requests is actual	2.7042 µs	0.004 µs	53.2124 µs	0.024 µs / 1457.81 µs[1]
Preprocessing is asymptotic	$O(1)$	$O(M^k)$	$O(\lvert K\rvert log\lvert K\rvert)$	$O(1)$

Source: **compiled by the authors**

[1] The mean time for processing one query using batch-prediction and one-by-one prediction accordingly

Moreover, some types of attacks may be committed deliberately, and they demand retaliatory measures for the maintenance of security.

Vote Omissions. Staying offline for a node can lead to a decrease in network performance. At the same time, committee members' votes can be absent for certain reasons. For example, an aggregator may not include them in its message, whether intentionally or not. In turn, the leader may not include an aggregated message in its block. It is not impossible to figure out exactly who is responsible for those omissions. However, we can assume that if a certain Coordinator misses voting several times in a row, this indicates its failure. Therefore, such a Coordinator should be penalized:

- a committee member does not vote $k = 4$ times in a row, not taking into account cases when aggregators do not deliver messages;
- a committee aggregator does not deliver messages $m = 2$ times in a row, not taking into account cases when slot leaders do not publish blocks.

In particular, this approach allows for constantly decreasing the share of Coordinators that stop working for an extended time. Otherwise, their growing number could significantly reduce the speed of block finalization.

All honest Coordinators make the decision to penalize faulty ones themselves, based on data from signed blocks when a corresponding omission series happens. The values of penalties equal $k \cdot v \cdot \alpha$ for a committee member, and $N \cdot m \cdot v \cdot \alpha$ for an aggregator, where v is taken from (1) and a scaling multiplier $\alpha \geq 1$. Hereinafter, the greater value of α makes the punishment more severe, so that the penalties are significantly higher than the potential harm caused to the network.

Missing Blocks. In the absence of previous block(s) in one or several slots in a row, the current slot leader refers to the last received block. The value of the penalty for the Coordinator that did not create a block is $C \cdot N \cdot v \cdot \alpha$. Hence, in both cases, the penalties equal the possible rewards for corresponding activities.

In addition, all penalized Workers in both cases mentioned above can no longer participate in the network functioning during the current Era and thenext one. In other words, they cannot be assigned as committee members or block producers from the following epoch through the end of the next Era. This is implemented to eliminate the causes of misbehavior, and to keep Workers' stakes from being sharply reduced when they are back in operation.

Duplicate Creation. According to protocol rules, the current leader must create only one block per slot in the Coordinating network. A Coordinator who discovers two blocks created in the same slot attaches them as proof when it is its turn to produce a block and receives 50 % of the penalty amount as a whistleblower reward.

Therefore, there is no need for further action by Coordinators to be generally agreed upon, and such rewards do not lead to inflation because all penalties are burned.

The value of $C \cdot N \cdot v \cdot \alpha$ is charged immediately from the faulty block producer. Hence, that leader loses its reward, since one of two blocks was previously included in the blockchain and the corresponding reward has already been paid. However, if there are n conflicting blocks, then the penalty equals $C \cdot N \cdot (n-1) \cdot v \cdot \alpha$. Proofs can be provided by different Coordinators, but they must contain no more than one of the conflicting blocks previously mentioned.

Conflicting Messages. A committee member may sign and send messages containing conflicting information (e.g. double voting in the same slot). When it is revealed, these messages are attached as proof by a whistleblower, and the penalty of $N \cdot v \cdot \alpha$ is charged to protect the network from spamming, since they could be sent to all committee members. In doing so, all actions are similar to the block duplicate creation case. Penalties are cumulative as well, and equal $(n-1) \cdot N \cdot v \cdot \alpha$ in general. For example, if there are three conflicting messages, then the penalty is doubled.

Invalid Proof. A leader may submit invalid proof of attacks within its block. Clearly, neither penalties nor rewards are charged, but another Coordinator may report this behavior by providing a reference to such a block. In this case, the penalty value applied to that leader is equal to double its possible benefit with the current v. For example, if an invalid proof reports two conflicting blocks, then the penalty will be $C \cdot N \cdot v \cdot \alpha$. In doing so, each Worker independently decides whether a proof is valid.

Proofs submitted repeatedly will not be executed. In other words, one cannot be penalized twice for the same attacks. In addition, the provision of such repeated proofs is an attack in itself, and is penalized as an invalid proof as well.

Attacks in the Shard Network

A Validator is entitled to create one block with transactions in a slot. If it releases more than one block in the same slot of the Shard network and those blocks are finalized in the Coordinating network, such a Validator unduly receives an additional benefit. Proof of this attack is two headers of the conflicting blocks signed by the malevolent Validator. Coordinators act similarly to the duplicate creation case in the Coordinating Network, but the penalty amount consists of all profits obtained from these blocks, multiplied by α.

Unlike block producing in the Coordinating network, a Validator can miss its turn to create a block in the Shard network without any penalty, but they lose any possible profit. This will not significantly affect the network performance, since several blocks are produced per slot by other Validators, and missed transactions will be published in the next slot. Moreover, if a Validator does not have time to synchronize before producing its block and refers to the old tip-blocks, its reward can be reduced appropriately, as mentioned above.

P2P Communication. Some types of attacks are committed during peer-to-peer (P2P) ([28], [29]) interactions and cannot be recorded in the ledger, e.g. spreading an invalid block. Hence, to ensure robust operation, each node should apply its own local communication management while building the network graph. Prioritization of communication with well-behaved nodes helps to reduce the processing load, obtain up-to-date information, and act in a timely manner within the consensus protocol with other nodes.

There are multiple approaches to implementing a local reputation system into a decentralized network, and almost all public P2P networks need to protect themselves from malicious activities like spreading invalid or unexpected information, spamming, deliberate delays in work, etc. For example, a node can inform others about a new ledger status to be synchronized but not send new blocks, suspending the work. Node software analyzing all incoming messages can reveal some types of misbehavior and stop it for a while, or even entirely block communication with such hostile nodes in the future.

CONCLUSIONS

The developed system of incentives is consistent with the Waterfall consensus to achieve a self-sustaining and high-performing network by incentivizing Workers' behaviors. However, the proposed mechanisms can be modified for a wide range of PoS consensus cases, depending on their distinct features, due to a flexible and transparent architecture, as well as a set of tuned parameters. The core principle is a fair reward distribution for well-behaved nodes and corresponding penalties for faulty nodes, to ensure a general economic equilibrium. In doing so, all honest Workers come to common decisions on the contributions of one another, based directly on the consensus protocol work of the Coordinating ledger, and do not require supplementary interactions. When designing tokenomics, upper limits were set on the commission for placing transactions. All the results described were obtained under the condition of this constant constraint. We managed to design a system in which there would be no unlimited growth of transaction fees, but the economic feasibility of the functioning of Workers would be preserved.

In addition, the incentivizing system promotes appropriate protection from diverse types of attacks [30], [31] like Nothing-at-stake [32], Rich-get-richer, Sybil, and Splitting, etc. [33], as well as faulty actions that are not done intentionally, where some possible threats have certain features related to a DAG structure.

Therefore, we can conclude that all the tasks are solved and the purpose of the study is achieved. Future work will center on researching and simulating malicious activities to develop a multi-parameter configuration that optimizes network performance, reliability, and security.

REFERENCES

1. Saad, S. M. S. & Radzi, R. Z. R. M. "Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos)". *International Journal of Innovative Computing*. 2020. DOI: https://doi.org/10.11113/ijic.v10n2.272.

2. Gächter, S., Orzen, H., Renner E. & Starmer, C. "Are experimental economists prone to framing effects? A natural field experiment". *Journal of Economic Behavior & Organization*. 2009; 70 (3): 443–446. DOI: https://doi.org/10.1016/j.jebo.2007.11.003.

3. “BitMEX Research. Ethereum’s Proof of Stake System – Calculating Penalties & Rewards”. 2021. – Available from: https://blog.bitmex.com/ethereums-proof-of-stake-system-calculating-penalties-rewards. – [Accessed: July, 2022].

4. Lau, K. “Ethereum 2.0. An Introduction”. Crypto.com, 2020. 25 p.

5. “Polkadot Wiki. Staking”. 2021. – Available from: https://wiki.polkadot.network/docs/en/learn-staking. – [Accessed: July, 2022].

6. Unchained, C. “Cosmos Validator Economics – Bridging the Economic System of Old into the New Age of Blockchains”. Cosmos Blog. 2018. – Available from: https://blog.cosmos.network/economics-of-proof-ofstake-bridging-the-economic-system-of-old-into-the-new-age-ofblockchains-3f17824e91db. – [Accessed: July, 2022].

7. “IOTA Foundation. Staking Start”. IOTA Blog. 2021. – Available from: https://blog.iota.org/iota-staking-start. – [Accessed: July, 2022].

8. Iyer, K., & Dannen, C. “Crypto-economics and game theory. Building Games with Ethereum Smart Contracts”. *Apress Berkeley*. 2018. p. 129–141.

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”. Preprint arXiv:1902.10865. 2019. DOI: https://doi.org/10.48550/arXiv.1902.10865.

10. Chang, Z., Guo, W., Guo, X., Zhou, Z. & Ristaniemi, T. “Incentive mechanism for edge-computing-based blockchain”. *IEEE Transactions on Industrial Informatics*. 2020; 16 (11): 7105–7114. DOI: https://doi.org/10.3390/en13195213.

11. Motepalli, S. & Jacobsen, H. A. “Reward mechanism for blockchains using evolutionary game theory”. *IEEE 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. 2021. p. 217–224. DOI: https://doi.org/10.48550/arXiv.2104.05849.

12. Amoussou-Guenou, Y., Del Pozzo, A., Potop-Butucaru, M. & Tucci-Piergiovanni, S. “Correctness and fairness of tendermint-core blockchains”. Preprint arXiv:1805.08429. 2018. DOI: https://doi.org/10.48550/arXiv.1805.08429.

13. Mazurok, I., Pienko, V. & Leonchyk, Y. “Empowering fault-tolerant consensus algorithm by economic leverages”. *ICTERI Workshops*. 2019. p. 465–472.

14. “Waterfall Foundation”. 2022. – Available from: https://waterfall.foundation/ – [Accessed: July, 2022].

15. Cullen, A., Ferraro, P., King, C. & Shorten, R. “On the resilience of DAG-based distributed ledgers in IoT applications”. *IEEE Internet of Things Journal*. 2020. p. 7112–7122.

16. Benčić, F. M. & Žarko, I. P. “Distributed ledger technology: Blockchain compared to directed acyclic graph”. *IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 2018. p. 1569–1570. DOI: https://doi.org/10.48550/arXiv.1804.10013.

17. Masalskyi, R. “DAG Distributed Ledger Modeling”. *The 1st Student Sci. Conf. of Joint Res.* Cooperation between Odessa I. I. Mechnikov National University and Huaiyin Institute of Technology. 2022. p. 171–175.

18. Kuipers, L. & Niederreiter, H. “Uniform distribution of sequences”. *Courier Corporation*. 2012.

19. Crow, E. L. & Shimizu, K. “Lognormal distributions”. New York: Marcel Dekker. 1987.

20. Ahsanullah, M., Kibria, B. M. & Shakil, M. “Normal distribution. Normal and student st distributions and their applications”. *Atlantis Press*. 2014. p. 7–50.

21. Yao, F. F. “A 3-space partition and its applications”. *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*. 1983. p. 258–263.

22. Dayhoff, J. E. “Neural network architectures: an introduction”. *Van Nostrand Reinhold Co.* 1990.

23. Arsirii, O. O. & Manikaeva, O. S. “Models and methods of intellectual analysis for medical-sociological monitoring’s data based on the neural network with a competitive layer”. *Applied Aspects of Information Technology*. 2019; 2 (3): 173–185. DOI: https://doi.org/19.15276/aait.03.2019.1.

24. Petrosiuk, D. V., Arsirii, O. O., Babilunha, O. Y. & Nikolenko, A. A. “Deep learning technology of convolutional neural networks for facial expression recognition”. *Applied Aspects of Information Technology*.2021; 4 (2): 192–201. DOI: https://doi.org/19.15276/aait.02.2021.6.

25.Cunningham, P., Cord, M. & Delany, S. J. “Supervised learning.” *Machine Learning Techniques for Multimedia*. *Springer*. 2008. p. 21–49.

26. Liu, B. “Supervised learning”. Web data mining. *Springer*. 2011. p. 63–132.

27. Niculescu-Mizil, A. & Caruana, R. “Predicting good probabilities with supervised learning”. *Proceedings of the 22nd International Conference on Machine Learning*. 2005. p. 625–632.

28. Schollmeier, R. “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”. *Proceedings First International Conference on Peer-to-Peer Computing*. 2001. p. 101–102.

29. Ford, B., Srisuresh, P. & Kegel, D. “Peer-to-Peer communication across network address translators”. *USENIX Annual Technical Conference*. 2005. p. 179–192.

30. Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D. & Mohaisen, D. “Exploring the attack surface of blockchain: A comprehensive survey”. *IEEE Communications Surveys & Tutorials*. 2020. p. 1977–2008.

31. Lin, I. C. & Liao, T. C. “A survey of blockchain security issues and challenges”. *Int. J. Netw. Secur*. 2017. p. 653–659.

32. Rose, D., Machery, E., Stich, S., Alai, M., Angelucci & A., Berniūnas. “Nothing at stake in knowledge”. *Noûs*. 2019. p. 224–247.

33. Mazurok, I. E., Leonchyk, Y. Y. & Kornylova, T. Y. “Proof-of-greed approach in the nxt consensus”. *Applied Aspects of Information Technology*. 2019; 2 (2): 153–160. DOI: https://doi.org/19.15276/aait.02.2019.6.

Conflicts of Interest: the authors declare no conflict of interest

Received 14.07.2022
Received after revision 12.09.2022
Accepted 11.10.2022

DOI: https://doi.org/10.15276/aait.05.2022.13
УДК 004.9

Система стимулювання для децентралізованих платформ на основі DAG

Мазурок Ігор Євгенович[1)]
ORCID: https://orcid.org/0000-0002-6658-5262; mazurok@onu.edu.ua. Scopus Author ID: 57210121184
Леончик Євген Юрійович[1)]
ORCID: https://orcid.org/0000-0003-1494-0741; leonchyk@onu.edu.ua. Scopus Author ID: 57192064365
Грибняк Сергій Сергійович[2)]
ORCID: https://orcid.org/0000-0001-6817-8057; s.s.grybniak@op.edu.ua
Нашиван Олександр Сергійович[2)]
ORCID: https://orcid.org/0000-0001-8281-4849; o.nashyvan@op.edu.ua
Масальський Руслан Олександрович[1)]
ORCID: https://orcid.org/0000-0002-8573-9802; masalskyi@stud.onu.edu.ua
[1)] Одеський національний університет ім. І. І. Мечнікова, вул. Дворянська, 2. Одеса, 65082, Україна
[2)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Децентралізовані публічні платформи стають все більш популярними через зростання кількості до-датків для різних сфер бізнесу, фінансів і соціального життя. Неавторизовані вузли можуть легко приєднатися до таких мереж без будь-якого підтвердження, що робить прозору систему винагород і покарань вирішальною для самодостатності публічних платформ. Щоб досягти цього, система заохо-чення та покарання за поведінку працівників має бути тісно узгоджена з відповідним консенсусним протоколом, враховуючи всі його особливості та сприяючи сприятливому середовищу з рівними пра-вами для всіх учасників. Основна мета винагород полягає в тому, щоб заохотити працівників дотри-муватись протоколу належним чином і покарати їх за будь-який тип неналежної поведінки. Питання винагороди та покарання за блоки в децентралізованих мережах добре вивчені, але структура поси-лань DAG розподіленого леджера змушує нас розробляти методи, які є більш актуальними. Оскільки структури посилань не можуть бути надійно підтверджені через те, що вони побудовані на основі миттєвої видимості блоків певним вузлом, ми пропонуємо встановлювати винагороди для блоків у мережі DAG на основі ступеня довіри топологічних структур. При цьому всі чесні вузли приймають спільні рішення лише на основі інформації, записаної в леджер, не перевантажуючи мережу додатко-вими взаємодіями, оскільки такі дані завжди ідентичні та доступні. Основною метою цієї роботи є розробка справедливого розподілу винагород серед чесних працівників та встановлення розмірів штрафів для винних, щоб забезпечити загальну економічну рівновагу платформи Waterfall. Запропонований підхід має гнучку та прозору архітектуру, що дозво-ляє використовувати такий підхід до широкого спектру консенсусних протоколів на основі PoS. Ос-новні принципи полягають у тому, що винагорода працівників залежить від важливості виконаної роботи для створення блоку та досягнення консенсусу, а їхні штрафи не повинні бути меншими за потенційний прибуток від можливих атак. Система стимулювання може полегшити захист від різних типів атак, а саме так званих атак Nothing at-stake, Rich-get-richer, Sybil і Splitting, а також від деяких конкретних загроз, пов’язаних зі структурою DAG.

Ключові слова: токеноміка; стимулювання; блокчейн; спрямований ациклічний граф; протокол консенсусу

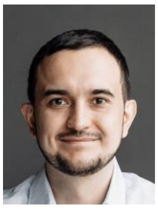
ABOUT THE AUTHORS

Igor Y. Mazurok - PhD in Engineering Sciences. Associate Prof. of the Department of Optimal Control and Economic Cybernetics, Odessa I. I. Mechnikov National University. 2, Dvoryanskaya Str. Odessa, 65082, Ukraine
ORCID: https://orcid.org/0000-0002-6658-5262; igor@mazurok.com. Scopus Author ID: 57210121184
Research field: Distributed computing; decentralized system design and modeling; artificial intelligence

Мазурок Ігор Євгенович - кандидат технічних наук. Доцент кафедри Оптимального керування та економічної кібернетики. Одеський національний університет ім. І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна

Yevhen Y. Leonchyk - PhD in Physics and Mathematics. Associate Prof. of the Department of Mathematical Analysis. Odessa I. I. Mechnikov National University. 2, Dvoryanskaya Str. Odessa, 65082, Ukraine
ORCID: https://orcid.org/0000-0003-1494-0741; leonchik@ukr.net. Scopus Author ID: 57192064365
Research field: Mathematical modeling of compute; environmental and economic complex systems; bblockchain technology

Леончик Євген Юрійович - кандидат фізико-математичних наук. Доцент кафедри Математичного аналізу. Одеський національний університет ім. І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна



Sergii S. Grybniak - PhD Student in Applied Mathematics and Information Technologies. Odessa Polytechnic National University. 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0001-6817-8057; s.s.grybniak@op.edu.ua
Research field: Blockchain and directed acyclic graph technology; distributed ledger technology; data science; decentralized systems design and governance models

Грибняк Сергій Сергійович - аспірант кафедри Прикладної математики та інформаційних технологій. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Oleksandr S. Nashyvan - Master of Software for Automated Systems. Odessa I. I. Mechnikov National University. 2, Dvoryanskaya Str. Odessa, 65082, Ukraine
ORCID: https://orcid.org/0000-0001-8281-4849; o.nashyvan@op.edu.ua
Research field: Software development; decentralized systems design; blockchain and directed acyclic graph technology

Нашиван Олександр Сергійович - Магістр программного обеспечения для автоматизированных систем. Одеський національний університет ім. І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна

Ruslan O. Masalskyi - Bachelor of Applied Mathematics, Master Student. Odessa I. I. Mechnikov National University. 2, Dvoryanskaya Str. Odessa, 65082, Ukraine
ORCID: https://orcid.org/0000-0002-8573-9802; masalskyi@stud.onu.edu.ua
Research field: Machine learning; blockchain and directed acyclic graph technology

Масальський Руслан Олександрович – магістрант. Одеський національний університет ім. І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна