An Incentive System for Decentralized DAG-based Platforms

Sergii Grybniak[1], s.s.grybniak@op.edu.ua Yevhen Leonchyk[2], leonchyk@onu.edu.ua
Ruslan Masalskyi[2], masalskyi@stud.onu.edu.ua Igor Mazurok[2], mazurok@onu.edu.ua
Oleksandr Nashyvan[1], o.nashyvan@op.edu.ua

1. Institute of Computer Systems, Odesa Polytechnic State University, Ukraine
2. Faculty of Mathematics, Physics and Information Technologies, Odesa I.I. Mechnikov National University, 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 integrated into the corresponding consensus protocol, taking into account all of its features, and facilitating a favorable and supportive environment with equal rights for all participants. 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 to establish values for penalties for faulty ones, to ensure the general economic equilibrium of the Waterfall platform.

Keywords – tokenomics, incentivizing, blockchain, directed acyclic graph, consensus protocol.

Introduction. In [1], the core principles of Waterfall's tokenomics were presented for achieving a sustainable, secure, and high-performing network, by driving behaviors of all participants with economic leverages. However, the issues of creating a fair distribution of rewards among platform Workers and setting values of penalties were not addressed in detail. This work deals with the incentivization of nodes to honestly perform their duties. Despite being a direct sequel to [1], 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) consensus protocols of decentralized networks.

The incentive mechanism is the backbone of any tokenomics. 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.

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 since their entire security relies on a staking mechanism. 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.

Related Works. The issue of incentivizing blockchain Validators is actively discussed by game theory researchers (e.g. [2], [3]). Some methods propose frameworks that could be applied to many PoW and PoS blockchains ([4], [5]) while some methods are tightly integrated into certain types of consensus ([6], [7]). 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.

Platform Overview. Waterfall 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 core, 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) with rapid finality Proof-of-Stake (PoS) consensus.

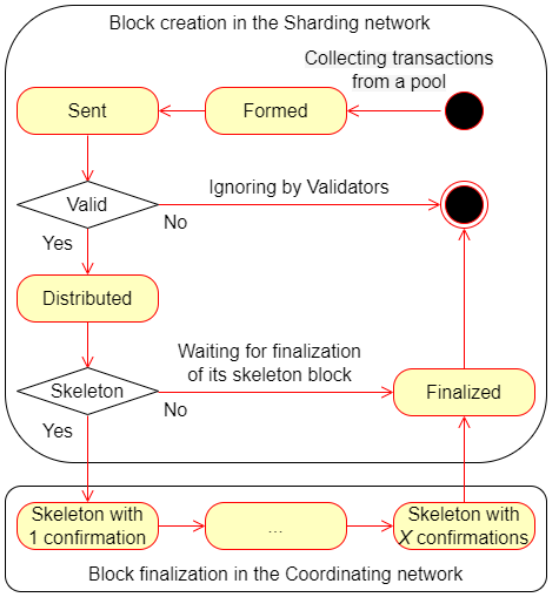
The platform consists of Coordinating and Sharding networks that achieve high transaction throughput via parallelized block production, thanks to the DAG structure. Each Worker consists of two parts, a Coordinator and a Validator, presenting it in corresponding networks. The timeline is divided into 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 Sharding 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. If the block is a skeleton in its slot, Validors send its hash to the Coordinating Network to be finalized. Otherwise, the block waits until another skeleton block is created in a future slot and links to it to be finalized. It should be noted that there is only one skeleton block per slot, and each of them must gain a few confirmations in the Coordinating Network to be finally accepted.

Rewards in the 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 Sharding 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.



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. A leader and aggregators are chosen from among ordinary committee members in every Era's slot. Note that an Era consists of a certain number of slots that each Coordinator is entitled to work as a slot leader.

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
$$v = \frac{W}{3 \cdot C \cdot N}$$
in case its vote message is included in a block of the Corresponding network. It should be noted that the value of v will be further used to define penalties.
2. Each of C aggregators can get
$$\frac{W}{3 \cdot C \cdot N} + \frac{W}{3 \cdot C} \cdot \gamma_1$$
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 this sum is received by the aggregator for work as an ordinary committee member.
3. Finally, a slot leader is rewarded by
$$\frac{W}{3 \cdot C \cdot N} + \frac{W}{3} \cdot \gamma_2$$

where $\gamma_2 \in (0,1]$ is a ratio of included aggregators' messages.

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

Rewards in the DAG-based Sharding network. The base block transaction fee f is split into two portions with a burning multiplier $l \in [l_0; 1]$ [1]:
$$f = l \cdot f + (1 - l) \cdot f$$
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 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 o f 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 [8]. As a result of modeling, a set of vectors was obtained:

$$B = \{\underline{b} = (b_1, b_2, \dots, b_k) : b_i \ - the\ number\ of\ references\ with\ depth\ i\ \} \subseteq \aleph^k$$

each of which corresponds to a block created in the Sharding network. Further, the following histogram g was generated:

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

that for each vector $\underline{b} \in B$ 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, the function $g(\underline{b})$ is normalized:

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

where $g_{max} = \max\limits_{\forall \underline{b} \in B} g(\underline{b})$. Then for each produced block $\underline{b}$ we can define the confidence function:

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

where $\underline{x} \le \underline{b} \Leftrightarrow \forall\, i\colon 1 < i \le k, x_i \le b_i$. The Validator's reward per block is determined in proportion to the degree of confidence, 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:

$$l \ = l_0 + (1 - l_0) \cdot (\ 1 - \ p(\underline{b}))$$

for each block, depending on its referential structure $\underline{b}$.

Attacks. In this chapter, 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]. 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 a scaling multiplier $\alpha \ge 1$. 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 the next 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 Sharding 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 Sharding 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 Sharding 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.

Conclusion. The developed system of incentives is integrated into 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.

In addition, the incentivizing system promotes appropriate protection from diverse types of attacks like Nothing-at-stake, Rich-get-richer, Sybil, and Splitting, etc, as well as faulty actions that are not done intentionally, where some possible threats have certain features related to a DAG structure. Future work will center on researching and simulating malicious activities to develop a multi-parameter configuration that optimizes network performance, reliability, and security.

1. S. Grybniak, Y. Leonchyk, R. Masalskyi, I. Mazurok, and O. Nashyvan, "Waterfall: Salto Collazo. Tokenomics," unpublished.
2. K. Iyer and C. Dannen, "Crypto-economics and game theory," Building Games with Ethereum Smart Contracts, Apress Berkeley, 2018, pp. 129–141.
3. Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A survey on applications of game theory in blockchain," preprint arXiv:1902.10865, 2019.
4. Z. Chang, W. Guo, X. Guo, Z. Zhou, and T. Ristaniemi, "Incentive mechanism for edge-computing-based blockchain," IEEE Transactions on Industrial Informatics, vol. 16 (11), 2020, pp. 7105–7114.
5. S. Motepalli, H. A. Jacobsen, "Reward mechanism for blockchains using evolutionary game theory," IEEE 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2021, pp. 217-224.
6. Y. Amoussou-Guenou, A. Del Pozzo, M. Potop-Butucaru, and S. Tucci-Piergiovanni, "Correctness and Fairness of Tendermint-core Blockchains," preprint arXiv:1805.08429, 2018.
7. I. Mazurok, V. Pienko, and Y. Leonchyk, "Empowering fault-tolerant consensus algorithm by economic leverages," ICTERI Workshops, 2019, pp. 465-472.
8. R. Masalskyi, "DAG Distributed Ledger Modeling," the 1st Student Sci. Conf. of Joint Res. Cooperation between Odesa I.I.Mechnikov National University and Huaiyin Institute of Technology, 2022, pp. 171–175.