










Probabilistic Optimization of Optimistic Finality for the Waterfall Consensus Protocol

Sergii Grybniak¹ , Yevhen Leonchyk² , Igor Mazurok² , Alisa Vorokhta²  , Oleksandr Nashyvan¹ , and Ruslan Shanin² 

¹ Odesa Polytechnic National University, Shevchenko Av. 1, Odesa 65044, Ukraine
sergii.grybniak@ieee.org

² Odesa I. I. Mechnykov National University, Dvoryans'ka St 2, Odesa 65082, Ukraine
alisa-vorokhta@stud.onu.edu.ua

Abstract. Blockchain is a distributed ledger technology that provides an immutable record and store of transactions. Today, one of the key challenges facing blockchain technology is the time required to finalize transactions. Mass adoption of payment systems and the development of enterprise-class decentralized systems have created a demand for a significant acceleration of finalization time in blockchains' networks, to facilitate fast and efficient transactions while maintaining security and performance. This article discusses the Waterfall platform, which is based on a Directed Acyclic Graph (DAG) architecture. Waterfall implements a two-level consensus protocol combining Ethereum's approach with a new algorithm that provides single-slot finality. However, the optimistic consensus involves a security trade-off that requires the maintenance of network scalability and performance. The proposed protocol modifications aim to minimize the time of transaction finality by obtaining an optimal level of blockchain Coordinators' support for slot finalization, building a simulation model for testing the modifications, and mitigating the problem of non-relayed transactions. The outcomes of this study will be incorporated into the Waterfall platform software, to enhance its dependability, efficiency, and security.

AQ1

AQ2

Keywords: Blockchain · Distributed Ledger Technology · Consensus Protocol · Transaction Finality · Positive Voting · Non-Relayed Transactions

1 Introduction

Blockchain technology is a secure decentralized system of digital record-keeping [1, 2]. System security mainly depends on the nodes' ability to finalize transactions through a consensus protocol. Finalization is the process of confirming transactions and adding them to the blockchain ledger, ensuring transaction validity and irreversibility [3]. The consensus mechanism determines how finalization is achieved. As blockchain technology continues to evolve, new consensus methods may be developed to improve the finalization process.

The Bitcoin network [4], the most well-known blockchain in the world, has a probabilistic finality concept – the more blocks added to the blockchain ledger after a transaction, the less likely the transaction is to be reversed. For example, it is believed that

the probability of a transaction being reversed after six added blocks is less than 0.1% [5].

On the Ethereum 2.0 blockchain, the finalized status of transactions is determined based on epochs [6]. Currently, the finalizing process takes about 2–3 epochs or 13–18 min. The reasoning behind this time duration was to strike a balance between decentralization, security, overall network load, customer expectations, etc. It is deemed appropriate for many use cases in various fields, including the decentralized finance (DeFi) industry [7]. However, in some business scenarios where many transactions are made, and for relatively small amounts, an average waiting time of approximately 15 min may not be acceptable. Note that there are 32 slots per epoch in Ethereum. Therefore, there are compelling arguments in favor of single-slot finality instead of epochs.

One possible solution to this problem is to develop an optimistic consensus protocol that ensures the specified probabilistic finality defined by a set of system parameters. This approach can significantly speed up the finalization of slots, and subsequently blocks themselves. However, to maintain network scalability and performance, a security trade-off must be made.

On the Waterfall platform [8], a two-level consensus protocol is implemented combining Ethereum's approach with a new algorithm that provides single-slot finalization, so that users have a choice. For example, figuratively speaking, if a user buys a cup of coffee, the deal may be promptly guided by the optimistic protocol, but more valuable arrangements should be made after waiting for 2–3 epochs.

The time required to finalize transactions also depends on how fast transactions are added to blocks from the pool, particularly their speed of propagation over the network and block occupancy. On the Waterfall platform, transactions are processed by Validators responsible for maintaining the integrity of Shard networks with DAG architecture [9]. As a result of the Shard working, the produced blocks of transactions are transmitted to network Coordinators for their linearization (ordering) and further finalization. However, there may be cases when Validators do not send received transactions to other Validators, leading to an uneven distribution of transactions between them. In addition, Coordinators may misbehave for various reasons, causing consensus delays.

The goal of our research is to enhance the consensus protocol “Waterfall: Gozalandia” [10] by minimizing the time of transaction finality. This will be accomplished through the completion of the following objectives:

- obtain the optimal level of blockchain Coordinators' support for slot finalization;
- build a simulation model for testing proposed protocol modifications;
- mitigate the problem of non-relayed transactions.

The study utilized various mathematical and statistical analysis techniques, as well as simulation modeling experiments in Python. The obtained outcomes will be incorporated into Waterfall platform software, which will enhance its dependability, efficiency, and security.

2 Literature Overview

The Ethereum consensus protocol has gained significant attention due to its recent transition to a new Proof-of-Stake (PoS) model [11], using a positive absolute supermajority rule that requires the support of two-thirds of total network stakeholders. This rule is intended to increase the security and decentralization of the blockchain network by ensuring that decisions are made with the agreement of a significant portion of network nodes. Despite the fact that Ethereum, as with any decentralized system, is not absolutely immune to attacks (e.g. [12, 13]), such an approach reveals itself as an effective governance mechanism in blockchain systems [14] and is constantly improving, including by cryptography and economic leverages.

The issue of how to finalize a single slot in PoS protocols is less researched in the literature than in other types of consensus protocols [15, 16]. The “Waterfall: Gozalandia” [10] presents an algorithm (so-called optimistic consensus) for single-slot finality, provided that epoch finalization is achieved in a timely manner. The main goal of this work is to acquire the optimal level of support from blockchain Coordinators accelerating the optimistic finalization of slots. In doing so, we consider Binomial distribution sampling that has been used in various fields to predict positive voting. Such an issue is well-studied in the case of two-alternative voting systems [18]. However, there are some distinctions in our case, since If a Coordinator has not positively voted, then it is considered faulty until it votes during the epoch.

In addition, efficient transaction dissemination is one of the crucial issues for transaction finality and is actively discussed in the blockchain research community. New relay protocols (e.g. [18, 19]) and incentivizing methods (e.g. [20–22]) are constantly emerging to enhance existing solutions and adapt them to new blockchain systems in accordance with their distinctive features. The Waterfall protocol also demands special modifications in the first place because of its DAG architecture, and significant increases in the number of Validators over the platform’s evolution.

3 General Platform Design

Waterfall is a decentralized network in which there is a set of independent Shards built on the blockDAG principle. There is also a separate Coordinating network, which is to finalize the sequences of transactions in Shards. A set of coordinated peer-to-peer independent software processes, called Workers, is created to implement such an architecture, consisting of two information-related components (see Fig. 1).

The first component, called the Validator, operates in the DAG network of a specific Shard and is responsible for creating and validating blocks there. The second component, called the Coordinator, works in a common Coordinating network and is responsible for linearization and intershard interactions. From a technical point of view, Workers are deployed on many physical nodes (servers), one or more on each server. Workers running on the same server have a common transaction pool, a common network state, and an archive. This reduces the cost of network deployment. Due to the large number of Workers, this technical solution does not negatively affect the degree of decentralization of the system.

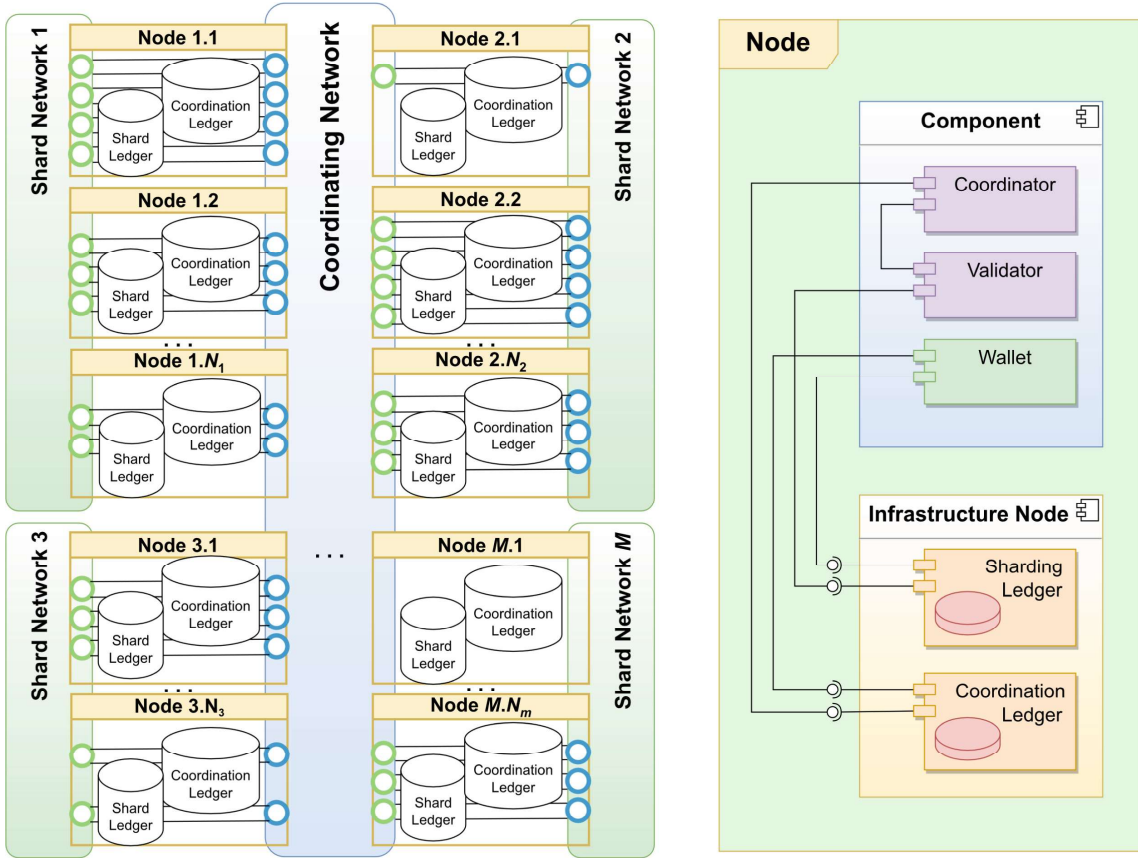


Fig. 1. Coordinating and Shard networks (the left panel) and the structure of a node (the right panel).

During the operation of the Coordination Network, all blockchain Coordinators are divided into 32 disjoint sets, according to the slots in which they work. In each slot, one of the Coordinators has the exclusive right to propose a block containing a list of DAG blocks with transactions to be finalized, and other Coordinators from this slot or subsequent slots can vote in support of this block. For the purposes of this paper, it is enough to know that slot results and, hence, the corresponding set of DAG blocks, are optimistically finalized if they have gained support in k – not necessarily consecutive – slots. In this work, we research a rule according to which a slot will be considered positively voted.

Some Coordinators for some reason may not send their vote for accepting the results of the slot. Such faulty participants make up a certain share f of the total number and are randomly distributed over the epoch slots. While in reality validator failures can occur at any time and for a fairly short period of time, we will only consider failures that occurred at the time of voting. Thus, each Coordinator can validate once per epoch, and the share of such Validators is limited by the value of f . The PoS consensus that provides epoch finalization requires the supermajority rule $f < 1/3$.

4 Probabilistic Slot Optimization

4.1 Slot Supporting Threshold

The number of votes supporting slot results can be considered as a random variable having the Binomial distribution with parameters n Coordinators per slot and success probability $p = 1 - f$. Figure 2 depicts probabilities of positive vote numbers as an example, with $n = 64$ and $p = 2/3$. Let us set the threshold value at which the slot will be considered as supporting the solution, $t = 1/2$. At the same time, the probability that honest (not faulty) Coordinators Y will collect a sufficient number of votes is equal to

$$P(Y > t \cdot n) = 1 - F_p(t \cdot n) = 0.9957, \quad (1)$$

where F_p is the Binomial cumulative distribution function. This probability will increase as n increases. For example, if the number of Coordinators per slot is doubled $n = 128$, it will exceed 0.9999. On the other hand, the threshold $t = 1/2$ will not allow multiple solutions to be supported in a slot, which would lead to an unresolvable contradiction in the system. However, an increase in the value of t seems to be inappropriate, e.g. the probability of honest voters reaching a ‘supermajority’ (with $t = 2/3$) is equal to 0.5235, which is on average 16.75 slots per epoch.

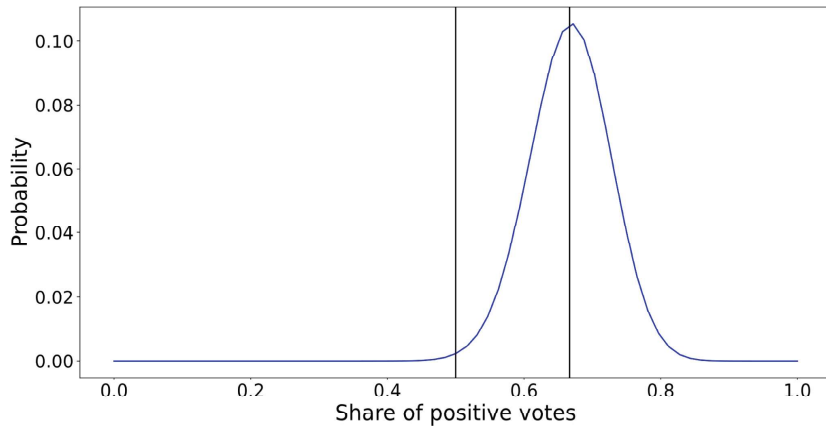


Fig. 2. The Binomial probability density function with $n = 64$ and $p = 2/3$.

From a practical point of view, the probability of reaching the majority by colluding malicious Coordinators within one slot is also of interest. As the most unfavorable case, we can assume that all the faulty participants ($f < 1/3$) are in collusion. Obviously, this is a complementary event to the abovementioned event, if we neglect the case when both groups (faulty and honest) get the same number of votes.

4.2 Simulating and Probabilistic Optimization

The considerations discussed above are applicable to the case when the share of all faulty network Coordinators f remains constant over slots. However, in practice, in the case of non-repetitive sampling (32 disjoint sets), the value of f may vary, although only slightly,

for a sufficiently large total number of voters. For testing, a voting model was built under the condition $f = 0.333$, which imitates the work of the Coordinating Network with a different number of n voters per slot for one million epochs. The finding presented in Table 1 is in line with theoretical results.

Table 1. Average values of positively voted slots.

$n =$	64	128	256	2,048	8,192
$t = 1/2$	31.885	31.998	31.999	32.0	32.0
$t = 2/3$	16.932	15.898	16.575	16.356	16.801

Figure 3 illustrates the general case for a fixed number of Coordinators $n = 256$ when the parameters f and t vary over a wide range. For other values of n , the graph shape remains the same, but as the number of voters increases, the jump becomes sharper. The simulation results confirm the expediency of choosing the majority rule with $t = 1/2$.

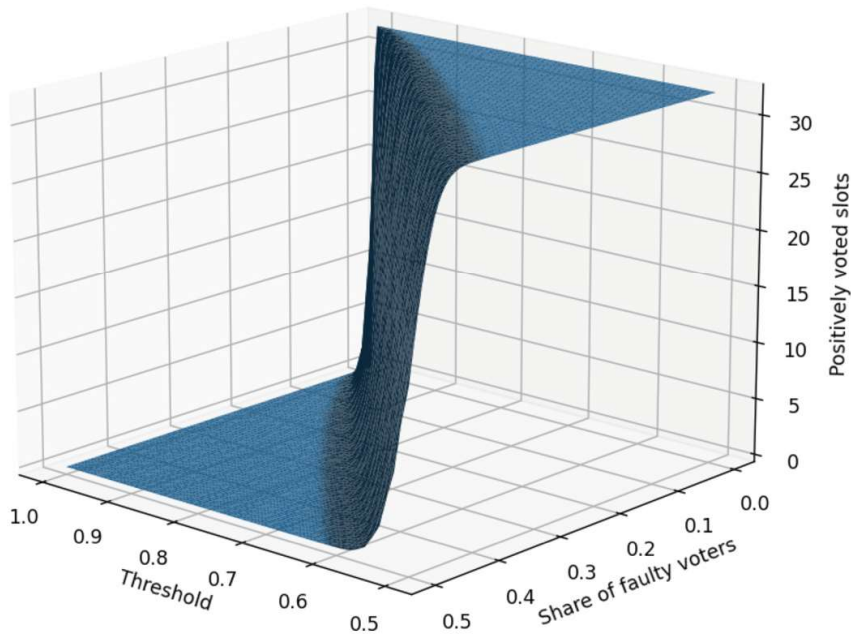


Fig. 3. Average values of positively voted slots with $n = 256$ Coordinators.

Let's assume that all the faulty Coordinators are in collusion and make a concerted effort to finalize a slot that is not going to be supported by the supermajority in the voting of an entire epoch. The attack begins when the leader of the slot is the Coordinator from among the conspirators that publishes its proposal for finalization. To validate this proposal (optimistic finalization), it is necessary that the conspirators are later able to gain control over the needed number of slots k (as a system parameter) faster than honest nodes that will support the "correct" competing proposal.

Here, as above, different numbers of Coordinators per slot acting during one million epochs were simulated. The shares of epochs when the group of conspirators with $f =$

1/3 was able to make a faulty optimistic consensus are presented in Table 2. With other values of n considered in Table 1, there was no faulty optimistic consensus. In all cases, the average numbers of optimistically finalized blocks in the Coordinating network were 20–21 per epoch, primarily due to faulty leaders (conspirators) producing faulty blocks that cannot be accepted by honest Coordinators. Therefore, one can recommend $k = 2$ for the Waterfall platform, since it will allow blocks to quickly reach optimistic finalization with a high confidence value.

Table 2. Shares of epochs with a faulty optimistic consensus.

$n =$	64	128
$k = 1$	0.029812	0.000439
$k = 2$	0.000314	–
$k = 3$	0.000002	–

5 Non-Relayed Transactions

Validators can withhold transactions for a variety of reasons. For example, Validators do not send transactions to others to include them in produced blocks on their own (so-called selfish mining), or Validators may be overwhelmed with incoming transactions (especially in high-throughput blockchain systems), causing them to drop some transactions to prioritize others. Non-relayed transactions can have significant consequences for the overall health of the network. In a supply chain management case, such transactions could result in delayed shipments or lost products; in a healthcare system, they could lead to delayed or inaccurate medical records, potentially endangering patient health, etc. As a whole, the problem of non-relayed transactions can have significant consequences for the security and efficiency of a blockchain network. Therefore, it is crucial for Validators to relay transactions promptly and efficiently, to ensure the integrity of the Shard network.

There are several approaches to prevent or at least to significantly reduce the appearance of non-related transactions in the Waterfall network:

- Due to the DAG structure, transactions are split into disjoint sets that must be processed by corresponding subnetworks without overloading specific nodes, even at peak times [23]. Thus, popular nodes receiving huge numbers of transactions will share with others those transactions that can be published only in other subnetworks.
- In the first stage, all nodes are located on cloud services and malicious owners cannot alter their software. Consequently, all Validators unconditionally follow the protocol [24] which was modified taking into account subnetworks, and transactions are propagated to all network nodes quickly and efficiently. Later, having a more significant number of Validators reduces the risk of non-related transactions caused by selfish mining or other malicious behavior.

- Burning the base transaction fee also reduces the attractiveness of selfish mining, but does not entirely eliminate it, since Validators get to keep tips for transaction publishing within the Waterfall tokenomics model [25].

One important problem with blockchain is the distribution of transactions and the replication of their pool. On some platforms, it is even considered good practice to not relay transactions from clients to other nodes. The transaction is “held” so that when creating a block, the node can include more transactions and receive a larger fee.

In Waterfall tokenomics, there is no significant direct economic interest for nodes to “hold” transactions in this way since fees are burned. Naturally, a node can get tips for hosted transactions, but the amount is not significant enough to justify breaking the protocol.

At the same time, it may be desirable to reduce synchronization traffic by transmitting the pool. Together, these two factors can stimulate changes in the software, which will lead to an increase in the publishing time of transactions received through faulty nodes that violate the protocol.

To eliminate the described problem in the Waterfall network, a mechanism of economic incentives is proposed, to reward compliance with the transaction distribution protocol. To accomplish this, each transaction transmitted from wallet to node is accompanied by an indication of the public key of the node through which the transaction is sent. Each wallet can transmit a transaction through several Entry Nodes at once. At the same time, the nonce of the transaction shows that it is the same transaction. However, different addresses of Entry Nodes make it possible to reward not only the producer of the block but also the Entry Node.

In this case, the reward is distributed as follows: $P = kP_i + (1 - k)P_b$, where P is the reward for publishing the transaction, P_i is the share of the reward of the block creator, P_b is the share of the reward of the Entry Node as the transaction provider. Currently $k = 0.5$, but the value of the distribution coefficient will be changed based on the results of the test network.

In addition to incentivizing compliance with the transaction distribution protocol, the described technical moment creates an additional point of responsibility if the receiving node signs the received transactions. The node that receives the transaction from the wallet checks the correctness of its address and, in case of an error and deliberate distortion, rejects the transaction.

In addition, a Validator’s monitoring tool could be added to enhance the current transaction dissimulation protocol. It will detect transactions pending for a long time, and well-behaving Validators will primarily include them in produced blocks.

6 Conclusion

Using methods of mathematical and statistical analysis, and after conducting a series of simulation experiments, we managed to reduce the transaction finalization time significantly and give early probabilistic estimates of the optimistic finalization of transaction blocks in Shards. The proposed solutions make it possible to obtain the optimal level of decision support in the Coordinating Network. The problem of speed of transaction

distribution and the presence of non-retransmitted transactions was mitigated by methods of economic incentives. Further work is aimed at designing a network monitoring tool for accurately estimating the likelihood of positive voting based on the results of the epoch, which can inform decision-making without additional assumptions on the value of f .

Currently, work is underway to implement the solutions obtained in the Waterfall platform software. The main elements of the protocol have been implemented, and load experiments are being conducted. Testing was carried out on t3.small and t3.medium AWS Servers with 2 cores CPU and 2 or 4 GB RAM respectively. According to preliminary data, this significantly increased the reliability, efficiency, and safety of decisions.

References

1. Sherman, A.T., Javani, F., Zhang, H., Golaszewski, E.: On the origins and variations of blockchain technologies. *IEEE Secur. Priv.* **17**(1), 72–77 (2019)
2. Bhutta, M.N.M., et al.: A survey on blockchain technology: evolution architecture and security. *IEEE Access* **9**, 61048–61073 (2021)
3. Anceaume, E., Pozzo, A., Rieutord, T., Tucci-Piergiovanni, S.: On finality in blockchains. [arXiv:2012.10172v1](https://arxiv.org/pdf/2012.10172.pdf) (2020). <https://arxiv.org/pdf/2012.10172.pdf>
4. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2009)
5. Bitcoin Wiki: Confirmation. <https://en.bitcoin.it/wiki/Confirmation>. Accessed 9 Mar 2023
6. Ethereum.org: Proof-of-stake (PoS) (2023). <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos>. Accessed 9 Mar 2023
7. Werner, S.M., Perez, D., Gudgeon, L., Klages-Mundt, A., Harz, D., Knottenbelt, W.J.: SoK: decentralized finance (DeFi). *arXiv preprint arXiv:2101.08778* (2021)
8. Waterfall: a Highly Scalable EVM-based Smart Contract Platform. <https://waterfall.foundation>. Accessed 9 Mar 2023
9. Grybniak, S., Dmytryshyn, D., Leonchyk, Y., Mazurok, I., Nashyvan, O., Shanin, R.: Waterfall: a scalable distributed ledger technology. In: *IEEE 1st GET Blockchain Forum*, California, United States (2022). In press
10. Grybniak, S.S., Leonchyk, Y.Y., Mazurok, I.Y., Nashyvan, O.S., Shanin, R.V.: Waterfall: Gozalandia. Distributed protocol with fast finality and proven safety and liveness. *IET Blockchain* 1–12, 465–472 (2023)
11. Lau, K.: *Ethereum 2.0. An Introduction* (2020)
12. Schwarz-Schilling, C., Neu, J., Monnot, B., Asgaonkar, A., Tas, E.N., Tse, D.: Three attacks on proof-of-stake ethereum. In: Eyal, I., Garay, J. (eds.) *Financial Cryptography and Data Security (FC 2022)*. LNCS, vol. 13411, pp. 560–576. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-18283-9_28
13. D’Amato, F., Neu, J., Tas, E.N., Tse, D.: No more attacks on proof-of-stake ethereum? *arXiv preprint arXiv:2209.03255* (2022)
14. Ethereum.org: Ethereum PoS Attack and Defense (2022). <https://ethereum.org/da/developers/docs/consensus-mechanisms/pos/attack-and-defense>. Accessed 9 Mar 2023
15. Buterin, V.: Paths toward single-slot finality. https://notes.ethereum.org/@vbuterin/single_slot_finality#Paths-toward-single-slot-finality. Accessed 9 Mar 2023
16. D’Amato, F., Zanolini, L.: A simple single slot finality protocol for ethereum. *arXiv preprint arXiv:2302.12745* (2023)
17. Mayfield, P.: Understanding binomial confidence intervals (1999). http://1989-6580.el-alt.com/binomial_confidence_interval.htm. Accessed 9 Mar 2023

18. Han, Y., Li, C., Li, P., Wu, M., Zhou, D., Long, F.: Shrec: bandwidth-efficient transaction relay in high-throughput blockchain systems. In: Proceedings of the 11th ACM Symposium on Cloud Computing, pp. 238–252 (2020)
19. Naumenko, G., Maxwell, G., Wuille, P., Fedorova, A., Beschastnikh, I.: Erelay: efficient transaction relay for bitcoin. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 817–831 (2019)
20. Wang, X., Chen, Y., Zhang, Q.: Incentivizing cooperative relay in UTXO-based blockchain network. *Comput. Netw.* **185**, 107631 (2021)
21. Zhang, J., Huang, Y.: TF: A blockchain system with incentivized transaction forwarding. In: IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), Bologna, Italy, pp. 213–223 (2022)
22. Babaioff, M., Dobzinski, S., Oren, S., Zohar, A.: On bitcoin and red balloons. In: The 13th ACM Conference on Electronic Commerce, pp. 56–73 (2012)
23. Antonenko, O., Grybniak, S., Guzey, D., Nashyvan, O., Shanin, R.: Subnetworks in BlockDAG. In: IEEE 1st GET Blockchain Forum, California, United States (2022)
24. Ethereum.org: Ethereum Wire Protocol (2022). <https://github.com/ethereum/devp2p/blob/master/caps/eth.md>. Accessed 9 Mar 2023
25. Grybniak, S., Leonchyk, Y., Masalskyi, R., Mazurok, I., Nashyvan, O.: Waterfall: Salto Col-lazo. Tokenomics. In: 2022 IEEE International Conference on Blockchain, Smart Healthcare and Emerging Technologies (SmartBlock4Health), Bucharest, Romania, pp. 1–6 (2022)