U°OS blockchain framework

January 25, 2019

Contents

Abstract

The concept of a cryptographically protected and distributed transaction ledger has demonstrated its efficiency in a series of projects. Decentralized frameworks based on blockchain technology allow communities to build transparent and reliable peer-to-peer systems that implement economic relationships between the users of the network. Over the recent years a series of consensus protocols were created and utilized in existing blockchain systems [4, 13, 14, 16, 17]. In this paper we introduce the U°OS blockchain framework and a novel DPoI (Delegated Proof of Importance) consensus algorithm that takes into consideration the value of the social and economic interactions between the members of the network and motivates users to actively contribute to the network growth. Notably, our DPoI metric can be modified to account for a variety of interactions, that arise in a particular network system. DPoI is a high-performance, resource efficient, network-growth inducing algorithm that rewards network participants for the economy enhancing operations in the system. Delegated Proof of Importance is an upgrade to the existing blockchain solutions, that integrates the concepts of the Delegated Proof of Stake (DPoS) and properties of networks. The system protocol is designed in accordance with business and end-user requirements such as privacy, transparency, and smooth volatility, which is achieved due to adaptive emissions proportionate to both network activity growth and the volume of the network itself, according to Metcalfe's law [1]. Our U°OS blockchain framework with the DPoI consensus metric represents a flexible architecture that can be deployed to set up a number of decentralized blockchain-based applications from social networks to service platforms with the direct growing economic value for all users. In this sense, U°OS is a unique framework that unlike any centralized system can be employed to create highly transparent network-based economies.

1 Introduction

An unprecedented increase in interest towards blockchain technologies has been observed in the recent years. At this time these technologies were primarily implemented in the form of distributed payment networks [4, 18, 19, 20] and distributed infrastructure systems [21, 22, 23, 24]. These payment networks are decentralized and enable fast low-cost peer-to-peer financial transactions between the users of the system, while infrastructure systems implement smart contracts and decentralized applications (DApps).

The basis of any blockchain system that also determines its technical characteristics is the network consensus algorithm. Namely, consensus algorithm is the mechanism that allows the network nodes to reach the agreement about the contents of the distributed ledger. In this paper we would like to review existing consensus protocol solutions and introduce a novel U°OS blockchain framework with a corresponding DPoI consensus metric.

1.1 Previous work

The problem of distributed consensus for networks with potentially fraudulent participants known as the Byzantine Generals' Problem was stated in 1982, long before the creation of blockchain. [2] Since then an array of different solutions has been developed [3]. However, the first solution that did not rely on a trusted third party, was the Proof-of-Work (PoW) algorithm [4]. Despite its advantages, PoW inherently has a number of shortcomings, namely, security [6], scalability, performance[5], the problem of progressive centralization of the networks around the largest mining pools [8] and, most importantly, the need to use vast volumes of physical resources, such as electricity and computing power to generate blocks. [9].

Computing resources needed for block hashing in the current blockchain frameworks, that implement PoW, are tremendous and far exceed the computing power of the world's greatest supercomputers. Energy use for the block mining is comparable to the power consumption of some countries and it continues to grow [7]. In 2012, in order to mitigate these shortcomings, the PPCoin, currently known as PeerCoin, cryptocurrency became the first to utilize an alternative consensus algorithm, Proof-of-Stake (PoS) [10]. In PoS consensus networks, the probability of creating a new block depends on the volume of tokens in a participant's account. Despite significant reduction in resource utilization, PoS turns out to have several drawbacks and in its current state, according to a number of experts, cannot serve as an adequate replacement to PoW [11], [12]. One of the major weaknesses of PoS is that

it additionally motivates users to concentrate all funds in one place or with one user, which leads to centralization of the network.

The next iteration of PoS was introduced as the Delegated Proof-of-Stake (DPoS) [13]. Here network members are divided into two groups: members, who delegate the authority to create blocks and validate transactions, and validators (block producers). This partition provides better scalability and efficiency. Nevertheless, DPoS still has the problem of motivation for a participant to use their assets actively instead of accumulating them, which has a negative impact on the growth and induces network centralization.

Yet another consensus metric, the Proof-of-Importance consensus algorithm (PoI), was first introduced in the NEM cryptocurrency [14]. PoI incentivizes network participants' activity. The major departure from PoS is that block generation probability and reward distribution depends not only on the volume of a user's deposits, but also on the participant's activity rate and reputation. Thus, the algorithm motivates users to be more active by participating in more transactions and contributing to the network development. Despite all its merits, PoI has some shortcomings in efficiency. [maybe good to have the citation here]

1.2 Motivation

As a matter of fact, social and economic interactions are integral components of any network system. Therefore, facilitation of social and economic activities contributes to the network development. We propose here a novel DPoI consensus algorithm that takes into account users' social and financial transactions in order to encourage participation in the network growth and prevent centralization. U°OS framework with DPoI consensus algorithm allows users to build virtually any network economy system on top of the U°OS blockchain and assign flexible financial and social activity scores, that reflect the nature of the socio-economic relationships in that particular system.

DPoI metric provides a mechanism for network growth via emission of tokens for social and economic transactions and also a mechanism of decentralization by accounting for socio-economic activity in the overall importance score. In a summary, DPoI metric supplies three main mechanisms for:

- network development facilitation via emission allocation for social and economic transactions
- decentralization via socio-economic importance score calculation
- deployment of DApps with a wide range of social and economic relationships

2 U°OS consensus algorithm

The major goal of U°OS project is to design a consensus algorithm with an individual influence score metric, that facilitates efficient score redistribution, motivates users to participate actively in the network development and prevents centralization. Modern blockchain solutions have problems with scalability, security and efficiency, and to solve those problems, U°OS protocol introduces the DPoI (Delegated Proof of Importance) Consensus Algorithm. This consensus algorithm combines the advantages of DPoS and PoI, and delegates validation rights to a limited number of accounts, based not only on the stake value of the protocol members, but also on the their transactional activity, in order to achieve high levels of efficiency and scalability within the network.

The U°OS consensus algorithm (Delegated Proof-of-Importance, DPoI) is based on the DPoS consensus algorithm [13]. In addition to the individual stake amount, our algorithm also considers incoming financial and social transactions of the user. In the U°OS Protocol participants have the option of delegating the right to validate blocks to a limited number of accounts through voting, using their personal importance scores, analogous to DPoS. Unlike DPoS, however, DPoI importance score formula is calculated from three components, namely, the stake amount, financial transfer activity and social activity. This framework is highly flexible since the network can collectively choose not only the weight of contribution of each term in the final importance score, but also decide on how to calculate the transfer and social activity scores, given the structure of economic and social interactions in the system. In general, the working principle of the U°OS Protocol Consensus Algorithm can be explained as follows.

2.1 Importance score calculation

DPoI importance score can be interpreted as an importance rating of an account i in the network. It is calculated using the formula below:

$$r_i = (1 - \omega_a - \omega_s)v_i + \omega_a\pi_i + \omega_s\sigma_i \tag{1}$$

where v_i is the stake volume index, π_i is the financial activity index, σ_i is the social network activity index, and ω_a and ω_s are the weight coefficients, that determine the relative significance of each component of the user activity.

Stake volume index is based on the amount of tokens owned and allocated by the account for the usage of the physical resources (CPU and bandwidth), and represents a balance proportion of the total amount of stake in the sys-

tem. Thus, an account with non-zero stake balance has non-zero importance score.

Social and financial activity indices depend on the transactional activity of the account in the most recent time window period, determined by the U°OS protocol. These indices are calculated using NCDAwareRank algorithm, described in the details in the sections below. We will characterize the main principles of the algorithm, based on the calculation of the financial index π_i. Later in the separate section we will explain how the algorithm is used to calculate the social activity component σ_i. NCDAwareRank gives more preference to the accounts that are tightly integrated in the general network, which helps to make the network resistant to the *splitting account attacks** performed by the botnets with a small number of accounts. Another important feature of the algorithm is the utilization of incoming activity only. This is a PageRank-based paradigm, that ensures a user can obtain the score only from the accounts that refer to it. Such a score is a direct representation of the user's utility for the entire network.

Financial activity index π_i is calculated only for the accounts with the stake balance exceeding the A_0 threshold. This value is determined by the network. When calculating the account financial activity index only transactions with the amount of tokens higher than T_0 are taken into account. That value is also determined by the network. Transactions that do not meet these thresholds are not included. Financial activity index depends on transactions, creation time of which lays in some pre-determined time interval. The duration of this interval is W blocks. Contribution of every transaction decreases exponentially. Comment for Lesha Prokopov: do we plan to use the same mechanism for the social activity index

Noticeably, each network application based on U°OS framework can define their unique set of financial and social transactions that are included in the calculation of the corresponding financial and social index scores. These applications can be deployed as DApps (distributed applications) and may implement a wide range of economic and/or social systems, such as service platforms, knowledge networks, digital content copyright systems, libraries, public records, online markets etc. Unlike centralized versions of such services, U°OS DApps would preserve the important properties of blockchain systems, particularly, decentralization, record immutability, and transparency.

*See Appendix 1 Glossary for definition

2.1.1 Financial activity score calculation

The vector of financial indices $\boldsymbol{\pi}^{(j)}$, where j is the indicator of the iteration in the algorithm, is calculated according to the NCDAwareRank, following the recurrent relation:

$$\boldsymbol{\pi}^{(j+1)} = (\alpha \boldsymbol{O} + \beta \boldsymbol{M} + (1 - \alpha - \beta)\boldsymbol{E})\boldsymbol{\pi}^{(j)} \tag{2}$$

Here $\boldsymbol{\pi}^{(j)}$ is a vector of account importance index values. The vector is normalized, i.e. the sum of its elements is 1. $\boldsymbol{O}$ is the outlink matrix, $\boldsymbol{M}$ is the interlevel proximity matrix. See the definitions of the matrices below. Coefficients α and β are the weights that determine contributions of the matrices $\boldsymbol{O}$ and $\boldsymbol{M}$. Their sum must be less than 1. $\boldsymbol{E}$ is the teleportation matrix, added to ensure that the series is convergent. This matrix is defined as follows:

$$\boldsymbol{E} = \frac{1}{N}\boldsymbol{e}$$

where N is the number of the accounts and $\boldsymbol{e}$ is the matrix in which all the elements are equal to 1, the calculation continues until for some j the following condition is fulfilled:

$$\|\boldsymbol{\pi}^{(j+1)} - \boldsymbol{\pi}^{(j)}\| < \delta$$

Here $\|\cdot\|$ is the vector norm defined as the sum of its elements, δ is the predetermined calculation accuracy. As an initial approximation, a vector $\boldsymbol{\pi}^{(\boldsymbol{0})}$ with all the elements equal to $\frac{1}{N}$ can be used.

Outlink matrix calculation: The outlink matrix $\boldsymbol{O}$ is calculated as follows. First, the weight matrix is calculated:

$$w_{ij} = \sum_{k|j\to i, h_k \geq H_0 \wedge h_k \leq H_0 + W} \theta(a_k - T_0)\theta(s_i - A_0)\theta(s_j - A_0) a_k \exp\left(lnK[\frac{h_k}{D}]\right)$$

where a_k is the token sum of transaction k, h_k is the k-th transaction depth, i.e. the block order number from the current point, also known as a block height, K and D are transaction contribution decrease parameters, that define how much the contribution of each transaction decreases over time, θ is the standard Heaviside step function. The purpose of these parameters is that over every D number of blocks, created after the given transaction, the transaction contribution decreases by $w' = Kw$. The sum is taken over all the transactions of a deposit from the account i to the account j, depth of which lays between H_0 and $H_0 + W$. H_0 and W are the parameters, which

values are currently equal to $H_0 = 2419200$ and $W = 1000$. In this setting only, the transactions from the time gap, duration of which is W blocks, contribute to the financial activity index calculation. Thus, we obtain the coefficient of the incoming financial activity from the user j to the user i as follows:

$$\hat{o}_{ij} = \begin{cases} w_{ji} - w_{ij} & \text{if } w_{ji} - w_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

After that the matrix, that was obtained, is normalized so that the sum of elements in every column is equal to 1.

$$o_{ij} = \begin{cases} \frac{\hat{o}_{ij}}{\sum\limits_k \hat{o}_{kj}} & \text{if } \sum\limits_k \hat{o}_{kj} > 0, \\ 0 & \text{otherwise} \end{cases}$$

Outlink matrix $\boldsymbol{O}$ captures the structure of incoming financial transactions in the network graph.

Interlevel proximity matrix calculation: Let S be the set of all the accounts used for the financial activity index calculation. S is further divided into disjoint subsets A_i, called NCD (Nearly-Completely Departed) blocks. These blocks are obtained using the SCAN algorithm, described below. For the given account u, G_u is the set of all the accounts, for which the according member of the outlink matrix o_{uv} is greater than zero. Then the set χ_u of proximal accounts of u is obtained as:

$$\chi_u = \bigcup_{v \in \{u\} \cup G_u} A_{(v)}$$

The interlevel proximity matrix is defined as follows:

$$M_{vu} = \begin{cases} \frac{1}{N_u |A_{(v)}|} & \text{if } v \in \chi_u \ , \\ 0 & \text{otherwise.} \end{cases}$$

where N_u is the number of NCD-blocks in χ_u. Interlevel proximity matrix M represents the structure of connectedness within the network.

SCAN-based network partitioning: SCAN algorithm is used to partition network graph into clusters and to prevent activity imitation or fraud between several affiliated accounts [15]. An indirected graph $G = \{V, E\}$ has every vertex, representing a user, and every edge, representing a non-zero

element of the outlink matrix. The structure of the vertex v is the set of all the adjacent vertices:

$$\Gamma(v) = \{w \in V | (v, w) \in E\} \cup \{v\}$$

The structural similarity of two vertices can be defined as follows:

$$\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

The vertex ε-neighborhood is a set of vertices for which

$$N_\varepsilon(v) = \{w \in \Gamma(v) | \sigma(v, w) \geq \varepsilon\}$$

The *CORE* is a vertex for which the number of elements in the ε-neighborhood is more than μ.

$$CORE_{\varepsilon,\mu}(v) \Leftrightarrow |N_\varepsilon(v)| \geq \mu$$

Vertex w is directly structurally reachable from vertex v if

$$DirREACH(v, w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \vee w \in N_\varepsilon(v)$$

Vertex w is structurally reachable from vertex v if

$$REACH(v, w) \Leftrightarrow \exists v_1, ...v_n \in V \forall i \in \{1, ...n - 1\} DirREACH(v_i, v_{i+1})$$

Vertex v is structurally connected with vertex w if

$$CONNECT(v, w) \Leftrightarrow \exists u \in V REACH(u, v) \vee REACH(u, w)$$

A cluster is a subset of vertices structurally connected to each other. It is possible to show that every vertex can only belong to one cluster. A vertex can also belong to no cluster; in this case it can either be a hub if there are vertices belonging to two different clusters in its environment, otherwise it is an independent vertex.

2.1.2 Social activity score calculation

Comment: this section is rapidly changing. Lesha will be committing more stuff here this week.

U°OS blockchain is integrated with the social network. Every account may have social relations with any other account and may have some content

belonging to it. Accounts and content objects with relations between them form a graph. Every account and content object obtains rate depending on involving into social relations. We denote this rate as social network activity index. It is contrubute to importance index, as defined in the formula 1.

Social index calculation is based on following data:

The matrix $\boldsymbol{V}$ contains all information about upvotes:

$$V_{ik} = \begin{cases} e^{(lnK)[h/H]} & \text{if the account } k \text{ upvoted the content } i \text{ at the height } h, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix $\boldsymbol{P}$ contains all information about content owners:

$$P_{ik} = \begin{cases} 1 & \text{if the content } k \text{ belongs to the account } i, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix $\boldsymbol{R}$ contains all information about reposts:

$$R_{ik} = \begin{cases} 1 & \text{if the content } i \text{ is a repost of the account } k, \\ -1 & \text{if } i = k \text{ and the content } i \text{ is a repost,} \\ 0 & \text{otherwise.} \end{cases}$$

Besides these matrices, calculations are based on the stack vector $\boldsymbol{s}$ and a weight vector $\boldsymbol{w}$, containing priority values for every account. These vectors are normalized to 1:

$$\sum_i \sigma_i^a = 1, \sum_i \sigma_i^c = 1$$

Based on these data structures, we may caclulate social activity vectors for accounts $\boldsymbol{\sigma^a}$ and content $\boldsymbol{\sigma^c}$

We may obtain now the matrix $\boldsymbol{V'}$:

And the matrix $\boldsymbol{U}$:

$$\hat{U_{ik}} = max_n(P_{in}V'_{nk})$$

$$U_{ik} = \begin{cases} \frac{\hat{U}_{ik}}{\sum_n \hat{U}_{nk}} & \text{if } \sum_n \hat{U}_{nk} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

We may now use the PageRank algorithm to calculate $\boldsymbol{\sigma^a}$:

$$\boldsymbol{\sigma^a}^{(n+1)} = (\alpha \boldsymbol{U} + (1-\alpha)\boldsymbol{T})\boldsymbol{\sigma^a}^{(n)}$$

The teleportation matrix $\boldsymbol{T}$ contains information about initial weights of accounts:

$$\boldsymbol{T} = (\beta \boldsymbol{e} + (1 - \beta)\boldsymbol{w})\boldsymbol{e}^T$$

Here β is a coefficient $0 < \beta < 1$, $\boldsymbol{e}$ is unit vector.

Based on $\boldsymbol{\sigma^a}$, we may obtain $\boldsymbol{\sigma^c}$:

$$\boldsymbol{\sigma^c} = \boldsymbol{V'}\boldsymbol{\sigma^a}$$

2.2 Importance score usage: network governance and emission

Importance index r_i in the framework is used for two major purposes:

- emission calculation for each user i
- network governance through voting for validators (block producers) and *calculating nodes*[†]

First, it defines the amount of new tokens received by the account in case when the emission is positive. See section 3 for details.

Second, importance index defines the relative weight of the account during the voting for producers and for the *calculating nodes*[‡]. A single user can offer their candidacy for both groups. Voting allows the delegation of certain powers within the system to a limited number of the validation accounts. Producers, that own nodes which produce and verify blocks, as well as the *calculating node*[§] owners, are selected by voting by other members of the network. Candidates nominate themselves for a desired position and voting can be performed by other users at any time. Changes are recorded within a short period, according to the protocol routine.

Change of the blockchain algorithm parameters and the block validation procedure is performed using voting as well. Here users delegate these decisions to the *calculating nodes*[¶].

Comment: we might want to include here the details of implementation of the changes in the blockchain algorithm, achieved by the consensus on the *calculating nodes*[‖], and on the block production, achieved by consensus among validators.

[†]See Appendix 1 Glossary for definition
[‡]See Appendix 1 Glossary for definition
[§]See Appendix 1 Glossary for definition
[¶]See Appendix 1 Glossary for definition
[‖]See Appendix 1 Glossary for definition

2.3 Key features of the algorithm: resistance to attacks

Following mitigations are used in the project in order to reduce the risk of attacks:

- Only incoming transfers contribute to the importance index, which makes obtaining score from the existing accounts more difficult.
- There are stake thresholds for accounts, participating in the importance index calculation, and also the amount thresholds for transfers, which makes attacks more expensive for the attackers.
- NCDAwareRank algorithm makes the system more resistant to *splitting account attacks***. This algorithm gives preference to accounts, tightly integrated into the common network.
- Coefficients ω_a and ω_s have relatively small values, this makes contribution of activity index and social index to total importance index significantly smaller than the stake contribution.
- Usage of the decaying weights makes effect of any fraud activity temporary.

**See Appendix 1 Glossary for definition

3 Emission

3.1 U°OS token utilization

U°OS tokens constitute the core of our crypto-economy. They are used in the system in several ways:

- to allocate CPU and bandwidth resources via smart contracts, using staked token amounts. Only core U°OS tokens can be used for CPU and bandwidth resource allocation.
- to purchase other resources, such as RAM, and perform other forms of financial transfers via smart contracts using unstaked tokens. Potentially, many types of tokens can be used for this activities.
- to vote for block producers and *calculating nodes*[††]. The amount of staked tokens, owned by the account, contributes to the user's weight during the voting process.
- to receive dynamic emission and to increase the importance score. Amount of tokens staked by the account directly influences the amount of dynamic emission and importance received by the user.

Thus, staked core tokens are used for the resource allocation and play an important part in emission and importance calculation, while unstaked tokens can be used in direct transfers.

Emission amount at launch constitutes one billion of protocol tokens, distributed to the original network accounts to start the protocol. The U°OS project implements adaptive emission. The emission volume is calculated regularly, in a certain time interval, $t_0, t_1, ...t_i$, where $t_{i+1} = t_i + T$. The volume of emission depends on the network activity growth in the preceding time period T.

3.2 Network activity calculation for a period of time

In order to calculate the emission, we first need to calculate the network activity for the unit time period T, defined by the U°OS protocol. To begin, we calculate the matrix of weights, according to the formula:

$$w_{ij}(t_n) = \sum_{k|j\to i, t_k \in [t_{n-1}, t_n]} a_k$$

[††]See Appendix 1 Glossary for definition

Here a_k is the sum amount in k-th transaction, t_k is the time at which k-th transaction was created. Summation is performed for all the transactions transferring any amount from account j to account i and created at the time frame from t_{n-1} to t_n.

In fact, each matrix element w_{ij} represents a weight of connection between account i and account j in a given time frame. Next we need to calculate the matrix of connections l:

$$l_{ij}(t_n) = \begin{cases} 1 & \text{if } w_{ji}(t_n) - w_{ij}(t_n) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

We calculate the activity in a given time frame as:

$$A(t_n) = \sum_{i,j} l_{ij}(t_n)$$

In this way, activity is calculated as a number of connections between active accounts in a set timeframe.

3.3 Emission value calculation

Target emission value E_T depends on the network activity growth. It defines the upper bound of the aggregate amount of the emission, that is achievable with the following activity value A:

$$\Delta A(t_n) = A(t_n) - A_{max}(t_{n-1})$$

$$E_T(t_n) = \begin{cases} E_T(t_{n-1}) + K_E \Delta A(t_n), & \text{when } \Delta A(t_n) > 0, \\ E_T(t_{n-1}) & \text{otherwise} \end{cases}$$

Here K_E is a coefficient that defines the maximum value of the emission with the activity increased by 1. $A_{max}(t_{n-1})$ is the previous maximum value since the system launch:

$$A_{max}(t_{n-1}) = max\Big(A(t_i), t_i \in [t_0, t_{n-1}]\Big)$$

Emission value, which is issued at a certain time t_n, is defined by the formula:

$$E(t_n) = \lambda S(t_{n-1}) f\Big(\kappa \frac{E_T(t_n) - E_S(t_{n-1})}{\lambda S(t_{n-1})}\Big)$$

Here λ is the marginal growth of the token amount in the system S Comment: we need to change the notation here, S is used elsewhere per one

emission. It is defined through L, which specifies the marginal growth S in a year, expressed as a percentage:

$$\lambda = (1 + \frac{L}{100})^{1/N} - 1$$

Here N is the number of emission issues per year.

$f(x)$ - is a sigmoidal function. In the present implementation of the algorithm a hyperbolic tangent is used as this function.

κ is a coefficient between 0 and 1 and it defines the speed at which a full emission approaches the target emission E_T if the activity level remains the same over the long term.

Initial values of both E_T and E_S are zero:

$$E_T(t_0) = 0, E_S(t_0) = 0$$

Dynamic emission allocated to a user for their social and financial activities motivates the user to participate in the network development, thus, helping to achieve the overall network growth.

4 U°OS framework architecture

Here we describe the system's architecture. I'm working with Petya Kotegov and Jenya Konstantinov to complete this section. We want to create an easy-to-understand high level architecture using schematic and textual instruments, and create references to the lower-level system descritions on the github. My plan is to have this part finished by 5th of February, and incorporate the feedback with improvements by the 9th.

Appendix 1 Glossary

U°OS consensus algorithm (Delegated Proof-of-Importance, DPoI)

The consensus algorithm, which is based on a calculation of the importance index of an account, which in turn depends on Stake Volume Index and Activity index

Importance index

The importance index of an account is calculated as a function of Stake Volume Index and Activity index

Account

An entity represented by a tuple of pairs of keys (public + private), which is registered in a blockchain by an individual.

Producer

Accounts with the right to verify blocks. They must have a node.

Node

A P2P network node, which performs all the calculations in blockchain. A node belonging to a producer account (producer node), produces blocks.

Splitting account attack

References

[1] Metcalfe, B. (2013). Metcalfe's law after 40 years of ethernet. Computer, 46(12), 26-31. URL: http://ieeexplore.ieee.org/abstract/document/6636305/

[2] Lamport, L., Shostak, R., Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401. URL: https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Byzantine-Generals-Problem.pdf

[3] Castro, M., Liskov, B. (2002). Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems (TOCS), 20(4), 398-461. URL: https://dl.acm.org/citation.cfm?doid=571637.571640

[4] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. URL: https://bitcoin.org/bitcoin.pdf

[5] Croman, K. et al. (2016). On scaling decentralized blockchains. In International Conference on Financial Cryptography and Data Security (pp. 106-125). Springer, Berlin, Heidelberg. URL: http://www.comp.nus.edu.sg/ prateeks/papers/Bitcoin-scaling.pdf

[6] Eyal, I., Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security (pp. 436-454). Springer, Berlin, Heidelberg. URL: https://arxiv.org/pdf/1311.0243.pdf

[7] Bitcoin Energy Consumption Index. digiconomist.net. URL: https://digiconomist.net/bitcoin-energy-consumption

[8] Buterin, V. (2014). Mining Pool Centralization at Crisis Levels. URL: https://bitcoinmagazine.com/articles/mining-pool-centralization-crisis-levels-1389302892/

[9] Bentov, I., Gabizon, A., Mizrahi, A. (2016). Cryptocurrencies without proof of work. In International Conference on Financial Cryptography and Data Security (pp. 142-157). Springer, Berlin, Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-662-53357-4_10/

[10] King, S., Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. URL: https://peercoin.net/assets/paper/peercoin-paper.pdf

[11] Demeester, T. (2017). Critique of Buterin's A Proof of Stake Design Philosophy. URL: https://medium.com/@tuurdemeester/critique-of-buterins-a-proof-of-stake-design-philosophy-49fc9ebb36c6

[12] Poelstra, A. (2014). Distributed consensus from proof of stake is impossible. URL: https://download.wpsoftware.net/bitcoin/old-pos.pdf

[13] Dantheman. (2017). DPOS Consensus Algorithm - The Missing White Paper. URL: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

[14] NEM Technical Reference. Version 1.2.1. February 23, 2018 URL: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf

[15] Xiaowei Xu et al. (2007). SCAN: A Structural Clustering Algorithm for Networks. URL: http://www1.se.cuhk.edu.hk/ hcheng/seg5010/slides/p824-xu.pdf

[16] Zhang E., A Byzantine Fault Tolerance Algorithm for Blockchain. URL: https://docs.neo.org/en-us/basic/consensus/whitepaper.html

[17] VIVA White paper (2017). URL: https://s3.amazonaws.com/vivacoin/viva-white-paper-v-2-0.pdf

[18] David Mazieres, (2016), The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus. URL: https://www.stellar.org/papers/stellar-consensus-protocol.pdf

[19] Evan Duffield and Daniel Diaz, (2018), Dash: A Payments-Focused Cryptocurrency. URL: https://github.com/dashpay/dash/wiki/Whitepaper

[20] Colin LeMahieu, (2015), Nano: A Feeless Distributed Cryptocurrency Network. URL: https://nano.org/en/whitepaper

[21] A Next-Generation Smart Contract and Decentralized Application Platform (2018). URL: https://github.com/ethereum/wiki/wiki/White-Paper

[22] EOS.IO Technical White Paper v2 (2018). URL: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[23] An Introduction to Hyperledger, (2017), URL: https://github.com/hyperledger/hyperledgerwp/blob/master/paper.pdf

[24] NEO White Paper: Smart Economy, (2018). URL: https://docs.neo.org/en-us/whitepaper.html