ISSN 2224-087X. Електроніка та інформаційні технології. 2018. Випуск 9. С. 106–112
Electronics and information technologies. 2018. Issue 9. P. 106–112

УДК 336.744:004.056.4:004.056.2

PROS AND CONS OF CONSENSUS ALGORITHM PROOF OF STAKE. DIFFERENCE IN THE NETWORK SAFETY IN PROOF OF WORK AND PROOF OF STAKE

O. Vashchuk, R. Shuwar

Ivan Franko National University of Lviv,
50, Drahomanova St., 79005, Lviv, Ukraine
olexlem@i.ua

The consensus algorithm is a mechanism that allows you to protect the network against attacks. The work of the algorithm is to provide rules that act on the network members. Proof of Work is one of the consensus algorithms based on the calculation of a complex algorithmic problem. This algorithm requires significant computing power to maintain its performance and therefore is superfluous. An alternative algorithm - Proof of Stake does not require so many resources to maintain network performance, but has a number of shortcomings.

The article describes the main aspects of the work of consensus algorithms Proof of Work and Proof of Stake. Also described objectivity and the main requirements to the algorithms in terms of CAP theorem. The comparison between algorithms shows their vulnerabilities to attacks, the features of work and the strengths.

Key words: blockchain, cryptocurrency, consensus protocol, proof of work, proof of stake, PoW, PoS, mining, minting, Sybill attack, DDoS attack, blockchain fork, CAP theorem.

Introduction. Participating in the cryptocurrency network requires each node to having its own copy of the blockchain, which is synchronized with other participators [2]. It’s obvious that every cryptocurrency must provide a way to secure its blockchain against attacks. For example, an attacker may spend some money and then reverse the spending transaction by broadcasting his own version of the blockchain, which does not include this transaction. Because the network is distributed, users have no knowledge as to which version of the ledger is valid.

Proof of Work algorithm provides the security of the network in form of block mining. The main point of PoW is that every node which want to participate in mining has to solve a computationally difficult problem to ensure the validity of the newly mined block. Every new block give some amount of coins for miner. The protocol is fair in the sense that a miner with p fraction of the total computational power can create a block with the probability p. An attacker is required to solve the same tasks as the rest participants of the PoW-secured network.

In the PoW-secured system protocol the security of the network is supported by physically scarce resources:

- specialized hardware,
- electricity.

This makes PoW-secured systems inefficient from a resource standpoint [1]. To increase the profit, miners are compelled to continuously deploy more resources for mining. This makes the cost of an attack on PoW-secured system extremely high, but ecological unfriendliness of the PoW protocol has resulted in efforts to build similar consensus protocol that are much less resource intensive.

One possible implementation with security not based on expensive computations relies on proof of stake (PoS) algorithms. The idea is simple: instead of mining power, the probability to create a block and receive the reward is proportional to a user's ownership stake in the system. An individual stakeholder who has p fraction of the total number of coins in circulation creates a new block with p probability. Users with the highest stakes in the system have the most interest to maintain a secure network, as they will suffer the most if the reputation and price of the cryptocurrency would diminish because of the attacks. To mount a successful attack, an outside attacker would need to acquire most of the currency, which would be prohibitively expensive for a popular system.

A cryptocurrency system is a form of a distributed database, with copies belonging to infrastructure providers for the currency communicating via a peer-to-peer Internet protocol. In terms of the CAP (consistency, availability and partition-tolerance) theorem, cryptocurrency systems are available (every request receives a response) and partition-tolerant (the service still performs even if some nodes fail), but are not consistent [3]. From time to time, different users of the system will see different states of the system as current. In some cases, the inconsistency corresponds to the situation when a new block hash has been discovered but has not yet been relayed to all users of the system. To obtain eventual consistency, a sound consensus protocol should impose the following requirement:

A user who discovered a block should be encouraged to broadcast it over the network immediately and not hold it for himself.

In other cases, system inconsistency is caused by the blockchain splitting into several branches. There are various causes of blockchain branching (forking) [1].

- Two users discover new blocks at about the same time
- An attacker attempts to reverse completed transactions by forking the blockchain.

In order to discourage deliberate branching, a sound consensus protocol should add the following requirement:

A user should be discouraged from discovering blocks on top of intermediate chains. If there are a known block B' referencing the block B, there should be no reason to build on B.

In order for the system to be eventually consistent, its consensus protocol should satisfy the following third requirement:

One of the competing branches should take over all other branches in a reasonable amount of time.

We will use separate terms for discovering blocks using proof of work and proof of stake algorithms:

The process of solving a computational challenge imposed by a proof of work protocol is called (block) mining, the process of solving a computational challenge imposed by a proof of stake protocol is called (block) minting.

Proof of Work. Consider Bitcoin as an example of a cryptocurrency system secured with a proof of work algorithm. Each block in Bitcoin have two parts:

- block header of key parameters, including block creation time, reference to the previous block and the Merkle tree root of the block of transactions;
- block list of transactions.

To reference a specific block, its header is hashed twice with the SHA-256 function; the resulting integer value belongs to the interval $[0, 2^{256} - 1]$ [3]. Using a generic hashing function - hash(·) with a variable number of arguments and range [0, M] will be without binding to the particular algorithm.

The block reference is used in the proof of work protocol; in order for a block to be considered valid, its reference must not exceed a certain threshold [7]:

$$Hash\ (Block) \leq M/D \qquad (1)$$

where $D \in [1, M]$ is the target difficulty. The only way to find Block satisfying (1) iterate through all possible variables in the block header repeatedly. The higher the value of difficulty, the more iterations are needed to find a valid block; the expected number of operations is exactly difficulty.

The time period T(r) for a miner with hardware capable of performing k operations per second to find a valid block is distributed exponentially with the rate k/D [7]:

$$P\ \{\ T(k) \leq t\ \} = 1 - \mathrm{exp}(-kt/D)$$

Consider n Bitcoin miners with hash rates $k_1, k_2, ..., k_n$. The period of time to find a block T is equal to the minimum value of random variables $T(k_i)$ assuming that the miner publishes a found block and it reaches other miners immediately. According to the properties of the exponential distribution, T is also distributed exponentially [7]:

$$P\ \{\ T\ def = \mathrm{min}\ (T_1, \ldots, T_n) \leq t\ \} = 1 - \mathrm{exp}(\ -t\ D \sum_n{}^{i=1} k_i\ )$$

$$P\ \{\ T = T_i\ \} = k_i / \sum^{j=1}{}_n k_j$$

The last equation shows that the mining is fair: a miner with a share of mining power p has the same probability p to solve a block before other miners [2].

Proof of Stake. In proof of stake algorithms, inequality (1) is modified to depend on the user's ownership of the cryptocurrency and not on block properties [4]. Consider a user with address A and balance - *balance*(A). A commonly used proof of stake algorithm uses a conditions[9]

$$Hash(\ Hash(\ Block_{prev}),\ A,\ time) \leq balance(A)\ M\ /\ D \qquad (2)$$

where

- $Block_{prev}$ denotes the block the user is building on,
- time is the current UTC timestamp.

Unlike (1), the only variable that the user can change is the timestamp t in the left part of the equation (2). The address balance is locked by the protocol; e.g., the protocol may calculate the balance based on funds that did not move for a day. There are no expensive computations involved in the proof of stake. Together with an address A and a timestamp t satisfying (2), a user must provide a proof of ownership of the address. To achieve this, he must have a private key corresponding to the address A. The time to find a block for address A is exponentially distributed with rate bal(A)/D. Consequently, the (2) implementation of proof of stake is fair: the probability to generate a valid block is equal to the ratio of user's balance of funds to the total amount of currency in circulation. The time to find a block for the entire network is distributed exponentially with rate ∑a bal(a)/D.

Delegated Proof of Stake. Delegated proof of stake (DPoS) is a generic term describing an evolution of the basic PoS consensus protocols. Blocks are minted by a predetermined set of users of the system (delegates), who are rewarded for their duty and are punished for malicious behavior (such as participation in double-spending attacks). In DPoS algorithms, delegates participate in two separate processes:

- building a block of transactions;
- verifying the validity of the generated block by digitally signing it.

While a block is created by a single user, to be considered valid, it typically needs to be signed by more than one delegate. The list of users eligible for signing blocks is changed periodically using certain rules. The set of delegates for each block is typically small. In some DPoS versions, a delegate needs to show commitment by depositing his funds into a time-locked security account (which is confiscated in case of malicious behavior); this version of DPoS is often referred to as deposit-based proof of stake. Delegated proof of stake does not use the conditions of (2). The stake is factored into DPoS with one of the following methods:

- delegates may be elected based on their stake in the system;
- delegates may receive votes from all users of the system with voting power depending on a voter's stake;
- delgates' votes on valid blocks may have power proportional to the size of their security deposit.

Objectivity of Consensus Protocols.

One of the tasks of a consensus protocol is to provide a way for newcomers to determine the current state of the system based on information received from peer nodes. This task is not trivial as some of the nodes can belong to a party performing a Sybil attack [8].

A consensus protocol is objective if a new node can independently arrive to the same current state as the rest of the network based on solely protocol rules (e.g., a definition of the genesis block) and messages propagated across the system (e.g., a set of all blocks).

Proof of work consensus is an example of an objective protocol; as long as a new node is connected to at least one "honest" user, it will choose the valid blockchain, as it has higher cumulative computational difficulty. Proof of stake, on the other hand, is not objective. If provided the attacker's fork is long enough, difficulty within it is adjusted to reflect the situation in which only accounts controlled by the attacker are active; this allows the attacker to generate a chain longer than the valid blockchain, but require substantial computing power. While long-range forks would be rejected by existing users of the system (e.g., by introducing a rule that limits the length of a possible fork), newcomers without prior knowledge of the current state would still choose the attacker's blockchain.

A consensus protocol is weakly subjective if a node needs a recent state in addition to protocol rules and messages propagated across the system to independently determine the current state of the system [5].

In the case of proof of stake, if there is a rule that disallows forks with a branching point more than N blocks in the past, it suffices to read the contents of a block with depth N or less to reliably determine the current state of the system. A newcomer can access this block from a trusted source (e.g., a website dedicated to the currency inquestion).

Comparison between PoW and PoS. The chance of a successful double-spend attack decreases as a transaction gains confirmations in PoW-based currencies and depends on the amount of mining power an attacker possesses[1]. To decrease the risk of funds double spending, it is recommended to wait for a certain number of confirmations (e.g., 6). Additionally, there are mechanisms to decrease the risk in fast payments.

For both of consensuses - proof of work and proof of stake, the types of attacks are common. The purpose of DoS attack is to disrupt the normal work of the cryptocurrency network by flooding the nodes. Sybil attack disrupts the network by creating a number of misbehaving nodes. The susceptibility of the network to DoS and Sybil attacks also depends on the details of the network protocol. There are no reasons that would make PoS less susceptible to these types of attacks compared to PoW. Selfish mining is specific for proof of work consensus. In selfish mining, an attacker selectively reveals mined blocks in purpose to waste computational resources of honest miners. The attack is ineffective for PoS currencies because in block generation involved not expensive resources. On the other hand, there are no known cases that a selfish mining attack has been successfully performed in Bitcoin, and some research argues that the attack description is based on faulty assumptions.

For PoW consensus, a degree of susceptibility to attacks can be predicted simply based on the total hashrate of the system [3]. In the case of PoS systems, there is no equivalent measure of the network "health status":

- if a stake is distributed evenly among many users, the system is prone to attacks that are based on a blockchain fork;
- if there are users with large stakes, they can disrupt the operation of the network (e.g., by censoring transactions).

The vulnerability of proof of work and proof of stake consensus mechanisms to attack types

Attack type	Vulnerability		
	PoW	PoS	Delegated PoS
Short range attack (e.g., bribe)	–	+	–
Long range attack	–	+	+
Coin age accumulation attack	–	+/–	–
Pre computing attack	–	+	–
Denial of service	+	+	+
Sybil attack	+	+	+
Selfish mining	+/–	–	–

Currently, there are several digital currencies implementing some form of proof of stake consensus including Peercoin, Nxt, Novacoin, BlackCoin and BitShares [6]. However, pure proof of stake approaches poses substantial security threats that cannot be recreated in proof of work systems (including Bitcoin).

These problems are inherent to proof of stake algorithms, as proof of stake consensus is not anchored in the physical world (cf. with hashing equipment in proof of work) [4]. That is why virtually all of the currencies relying on proof of stake use additional mechanisms to address security issues. Unlike proof of work, proof of stake consensus is not objective; the state of a PoS system cannot be reliably determined by new users based solely on protocol rules and a list of blocks and other network messages obtained from peers. In order to prevent long-range forks of the blockchain, a proof of stake system needs to implement weak subjectivity by combining protocol rules with social-driven security[5]. The social component of PoS systems weakens their decentralization and mathematical soundness.

Recent developments in proof of stake are delegated systems. While these systems solve several major problems with the straightforward PoS implementations, they are not yet widespread, making it difficult to evaluate their security. Nevertheless, delegated PoS solves the "nothing at stake" problem and prevents short range attacks on the system [5].

Conclusions. According to the researches, by 2020 the amount of energy that will be needed to ensure the work of Bitcoin network (which is one of the largest networks using PoW) is the same as Denmark uses. Another important negative aspect is that the bills for electricity are paid using fiat money, which exerts additional pressure on the cryptocurrency exchange rate. Therefore, it's obvious that the PoW protocol is superfluous and needs to be replaced.

A consensus protocol proof of stake does not properly protect distributed systems, which can not be said about the consensus protocol proof of work (including implementation for Bitcoin). This problem is typical for proof of stake since the proof of stake is abstracted from the physical world and exists only in the system (on the other hand, this is what makes it attractive). In the end, when using the PoS, the validator must have funds in the currency, while POW miner does not necessarily need the currency. That is why all currencies based on proof of stake use additional mechanisms to solve security issues.

For example, the problem of initial distribution can be solved using a limited-time version of the proof of work. You can prevent a double cost attack by including information about the last blocks in the transaction. However, these improvements are incomplete. Unlike the proof of work, the proof of the stake is not objective; the state of the system cannot be reliably determined by new users, based solely on the protocol of the rules, the list of blocks and other network messages received from peers. Delegated systems are among recent developments in proof of stake. Although these systems solve several major problems arising from the use of the usual PoS, they are not yet widespread, which makes it difficult to assess their safety. Nevertheless, the delegated POS resolves the problem of "nothing at stake" and prevents short-term attacks on the system.

As you can see, none of the consensus algorithms, even using a huge amount of resources, does not provide absolute protection against all types of attacks. Each of them has weaknesses that put the network at risk. However, it is worth considering consensus algorithms only as tools for ensuring network stability. By combining protocols together, or by creating hybrid protocols, you can get significantly better results. Finally, from all the protocols discussed in the article, proof of work provides the network with the most reliability since it prevents the long-range attacks, unlike the delegated proof of stake.

REFERENCES

1. *Gervais A.* On the Security and Performance of Proof of Work Blockchains / A. Gervais, G. O. Karame, K. Wüst and others // CCS '16 Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. – 2016. – P. 3–16.
2. *Nakamoto S.* Bitcoin: A Peer-to-Peer Electronic Cash System [Electronic resource] / Satoshi Nakamoto // Resource access mode:https://pdos.csail.mit.edu/6.824/papers/bitcoin.pdf.
3. *Nayak K.* Stubborn mining: Generalizing selfish mining and combining with an eclipse attack / K. Nayak, S. Kumar, A. Miller, E. Shi. // IACR Cryptology ePrint Archive. – 2015.
4. QuantumMechanic. Proof of stake [Електронний ресурс] / QuantumMechanic. – 2011. – Режим доступу до ресурсу: https://bitcointalk.org/index.php?topic=27787.0.
5. *Buterin V.* Proof of stake: How I learned to love weak subjectivity. [Електронний ресурс] / Vitalik Buterin // Ethereum Blog – Режим доступу до ресурсу: https://blog.ethereum.org/2014/11/25/proof-stake-learned-love-weak-subjectivity

6. *Bentov I.* Cryptocurrencies without proof of work. / I. Bentov, A. Gabizon, A. Mizrahi. // Chapter, Financial Cryptography and Data Security: FC 2016 International Workshops. – 2016. – P. 142–157.
7. Difficulty calculation (Bitcoin) [Electronic resource] – Resource access mode: https://en.bitcoin.it/wiki/Difficulty.
8. *Levine B. A.* Survey of Solutions to the Sybil Attack [Electronic resource] / B. Levine, C. Shields, N. Margolin. – 2006. – Resource access mode: https://www.researchgate.net/publication/228339775_A_Survey_of_Solutions_to_the_Sybil_attack.
9. *Buterin V.* On Stake [Electronic resource] / Vitalik Buterin // Ethereum Blog. – 2014. – Resource access mode: https://blog.ethereum.org/2014/07/05/stake/.

Стаття: надійшла до редакції 16.04.2018,
доопрацьована 24.04.2018,
прийнята до друку 29.04.2018.

ПЛЮСИ ТА МІНУСИ АЛГОРИТМУ КОНСЕНСУСУ PROOF OF STAKE. ВІДМІННОСТІ В БЕЗПЕЦІ МЕРЕЖІ У PROOF OF WORK ТА PROOF OF STAKE

О. Ващук, Р. Шувар

Львівський національний університет імені Івана Франка,
Драгоманова, 50, 79005, Львів, Україна
olexlem@i.ua

Алгоритм консенсусу – це механізм, створений для захисту мережі від атак. Робота алгоритму полягає в забезпеченні правил, які діють на членів мережі і спрямовані на недопущення в мережу невалідних даних та регламентують дії в разі розщеплення блокчейну на декілька гілок. Proof of Work – це один з алгоритмів консенсусу на основі розрахунку складної алгоритмічної задачі. Цей алгоритм потребує значної обчислювальної потужності, щоб підтримувати його продуктивність, тому є надлишковим. Альтернативний Proof of Stake не потребує стільки обчислювальних ресурсів для підтримки мережі, ґрунтуючись на тому, що в разі виявлення обману знищує заставу зловмисника, проте має низку недоліків.

Одним із завдань алгоритму консенсусу є визначення стану системи відносно отриманої інформації від вузлів мережі. Така ситуація виникає тоді, коли в мережу приєднується новий вузол і потрібно захистити його від отримання неправдивої інформації. Proof of Work є зразком об’єктивного алгоритму консенсусу, а от Proof of Stake – слабко об’єктивного.

Описано основні аспекти роботи алгоритмів консенсусу Proof of Work та Proof of Stake і вимоги до них за теоремою CAP. Теорема CAP є важливим правилом у випадку проектування розподілених систем і відносно неї система криптовалюти є стійкою до розділення та доступною, проте не гарантує цілісності результату. Складене порівняння між алгоритмами засвідчує їхню уразливість до атак, таких як атака Сибілли, DDoS атака, егоїстичний видобуток монет, короткотермінові та довготермінові атаки. Розглянуто особливості роботи, зокрема, такі такі як постійне поліпшення обладнання та збільшення електроспоживання у Proof of Work, а також вплив віку монет та їхньої кількості на можливість підписувати блоки, вибір валідаторів у різновидах Proof of Stake.

Ключові слова: блокчейн, криптовалюта, консенсус протокол, доказ роботи, доказ частки, видобуток, монетництво, атака Сибілли, атака DDoS, вилка блокчейну, теорема CAP.