RESEARCH ARTICLE

Network Dispersity-based Consensus Protocol

Yj1190590*[†]

Abstract. This study describes a blockchain consensus protocol based on network terminals and transmission latency. The protocol provides a stake-voting mechanism that has low electric power cost and avoids the limitations of the Proof-of-Stake (PoS) protocol. The stake-voting mechanism helps build a scalable multi-chain structure that may influence the future direction of cryptocurrencies. Moreover, competition relative to the number of terminals and the degree of dispersion involves many developers as potential miners to maintain or expand the network.

KEY WORDS

1. Proof of Network Dispersity(PoND). 2. Vote. 3. Reward distribution. 4. Self-interest.

1. Introduction

The Proof-of-Work (PoW) protocol has been criticized due to its high consumption of electrical energy. Nevertheless, the PoW protocol is an essential and effective method for the blockchain system. With the logic of “able people should do more work,” PoW is the first consensus protocol that can be used to realize a completely decentralized system. Protocols, such as the Proof of Activity, Proof of Burn, Proof of Storage, and Proof of Elapsed Time, adopt the same logic as the PoW. The Proof-of-Stake (PoS) protocol is the most successful among all other protocols because it does not require external resources and consumes less energy by expanding the main logic from “able people should do more work” to “rich people should do more work.” This restricts mining competition to static inner states, thereby eliminating the need for high power consumption. However, PoS has certain limitations. For example, “Nothing at Stake” problem can cause a double voting problem and history attacks; the “rich people should do more work” logic inevitably causes a problem. Whenever Miners spend the same amount of time, richer Miners earn more; thus, they tend to spend more time working. Thus, rich miners continue to become richer; this process will continue until only the richest users remain in the system. Further, the wealth distribution obeys the Pareto’s law. This means most wealth is acquired by few users; therefore, the extent of robbing the poor to feed the rich would be more significant. Some PoS-like protocols offer “interests” to all stakeholders to replace the mining process. Such protocols resolve the wealth-concentration problem but are vulnerable because they lack incentives to maintain the network.

This study aims to determine a proof-of-ability protocol to replace the PoS protocol. As a greater number of dispersed online terminals work together, the faster they can acquire randomly distributed information in the network due to network latency. This type of “ability” is referred to as “network dispersity,” which cannot be improved by enhancing the performance of single machines or by using more electric power. The “stake” property plays an important role as a measurement in the protocol due to its unduplicatable feature. Users can freely maintain the network using their abilities or stakes. The power of mining will be balanced via a dynamic adjustment of the reward distribution to maximize fairness and safety.

[†]Yj1190590 (3171228@qq.com)

All consensus protocols discussed herein relate to fully decentralized systems, and protocols, such as PBFT, DPOS, and Ripple are not considered.

2. Scenario and Characters

The entire network can be considered as a type of canvassing scenario that involves three types of characters according to a node's functions.

Voter. Each node that broadcasts a transaction could be a voter. Voters are primarily responsible for responding to "canvass" requests from workers and publishing transactions along with their votes. A stake held by a voter is considered "votes" in this scenario, and the number of "votes" determines mining competition. Any user can be a voter.

Miner. Miners have workers that canvass for "votes" throughout the network, and they use these "votes" to compete in block generation. Any user can be a miner.

Worker. Workers detect nearby voters and send a "canvass" request as quickly as possible. Workers primarily run on network terminals by being embedded in client apps or web sources.

The network scheme is shown in Figure 1.

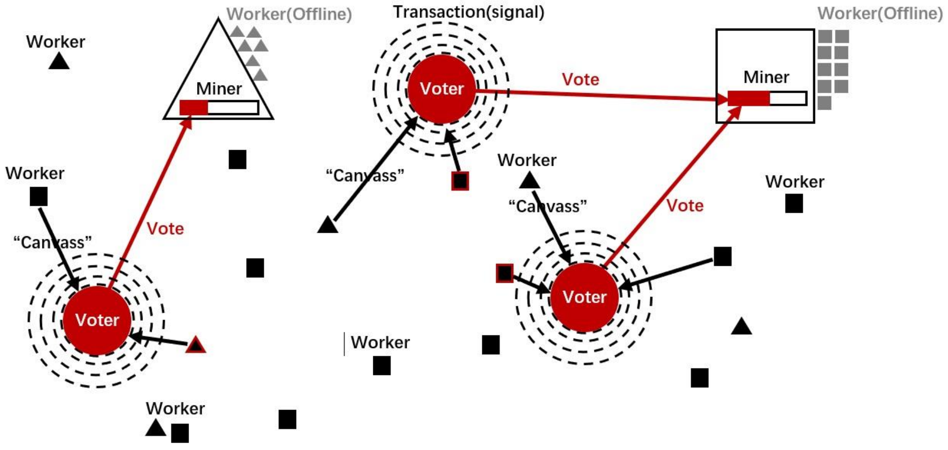


Fig. 1. Network scheme
Miners with more dispersed online workers have a greater likelihood of winning votes (i.e., stakes), which helps miners win mining competitions later.[1]

3. Consensus Process

The consensus process is divided into the following four stages.

3.1 Voting stage

A consensus is reached by conducting two types of voting: (1) the voter node votes on the miner's right to generate a block (recorded as x-vote) and (2) the miner node votes on the current main branch on behalf of the voter (recorded as y-vote) as follows:

(1) Before each transaction is released, the voter sends a broadcast signal and the worker nodes nearby send a canvass request to the voter after receiving the broadcast signal.

(2) After receiving the first request, the voter briefly communicates with the worker. Meanwhile, the worker obtains the block (recorded as b) on top of the main branch from the miner (recorded as m) and the miner's signature information and sends them to the voter.

(3) The voter writes the block hash (b), the miner’s account address (m), and signature information into the transaction structure and broadcasts the transaction.

(4) This transaction is confirmed by a miner who then packs it into a new block and publishes it on the Internet.

Each transaction will follow the abovementioned steps[2] to conduct both types of voting. To control the voting frequency, the voting ability owned by the account (record as x stake and y stake) must be adjusted separately. For x stake, voting interval of each account is considered as the adjustment coefficient. The stake starts at zero, and a certain proportion is increased every time the interval adds a time unit *t* (e.g., 10 blocks). The stake reaches its maximum value when the interval increases to a certain voting cycle (e.g., ~100 h for 6000 blocks). As for y stake, *t* is equal to the voting cycle, which means that the y stake comes into effect only once in an interval of the voting cycle. Similarly, each UTXO is added with x and y stakes and the transfer interval is used as the adjustment coefficient to adjust the number of voting stakes. The method and parameters are similar to those used for adjusting the corresponding account stakes.

3.2 Counting stage

In the counting stage, two types of voting are separately counted as follows:

(1) Equally distribute the x and y stakes of each voter in the block of the last voting cycle to each of their transactions in the block.

(2) Record the set of all transactions in the last voting cycle as T, check the b field of the transactions in set T, collect all the transactions of b == pre_hash (i.e., the previous block field in the blockhead)[3] and record them as set T', T'⊆T.

(3) Remove the relevant transactions of each miner before the last time of block generation from set T' and record the resulting set as set T'', T''⊆T'. [4]

(4) In set T'', collect the m transaction field and their assigned x stakes in Step (1) to obtain the number of x voting stakes acquired by each miner and record them as set X.

(5) In set T, collect the m transaction field and their assigned y stakes in Step (1) to obtain the number of y voting stakes acquired by each miner and record them as set Y.

By performing one-by-one statistics for the transactions in each block, we can obtain the number of x stakes obtained by all the miners involved in the competition and the number of y stakes of voters that the miners represent and complete the counting. The results are set X and set Y. For any time point, such as the counting of the block at position a, it refers to the statistical results one voting cycle ahead of block a, and are recorded as X_a and Y_a.

Figures 2 and 3 show the process of competing for block generation (x-vote) and the main branch (y-vote), respectively.

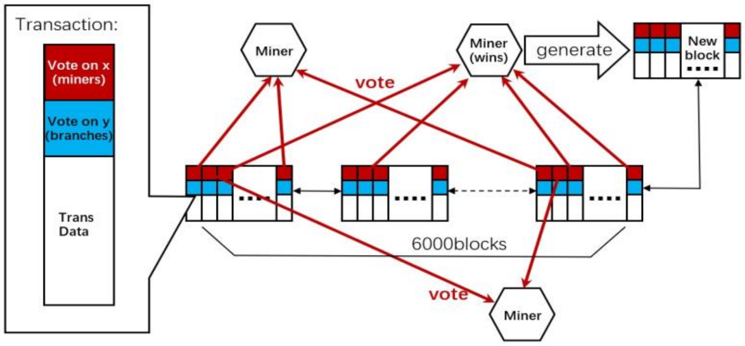


Fig. 2. Process of x-votes taking effect
The system counts votes in existing blocks, and the miner with the maximum number of votes has the greatest chance to win the competition.

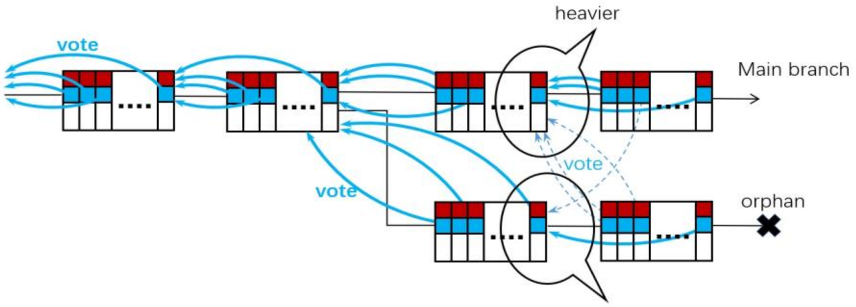


Fig. 3. Process of y-votes taking effect
When a fork occurs, voters vote between the branches and the votes are recorded in the next block. This determines the number of votes (stakes) for both branches in the next block generation. This also determines the weight of branch. The heaviest branch is the main branch.

3.3 Competition stage

The main task of the competition stage is to determine the miner that generates the block; the miner operates as follows:

(1) The miner performs mathematical operations based on constants such as timestamp and personal signature. If the expected result meets goals and requirements of block generation, it is recorded as hashProof()<target*d*x, where target refers to the goal; d refers to the difficulty adjustment parameter[5]; and x refers to the number of x stakes ($x \in X$) obtained by the current miner.

(2) After the requirements of block generation are met, the miner node packs the transactions received during this time period, generates the block, and posts it online. The earnings of each node and other calculation parameters are also packed simultaneously (transaction ID can be coded in 6000 block range to save space). The earnings obtained during mining are distributed pro rata to the miner and all the voters involved.

3.4 Confirmation stage

In PoND, the priority of branches does not depend on their length but depends on their weight[6]. The weight of a branch without furcation whose length is shorter than one voting cycle is calculated as follows:

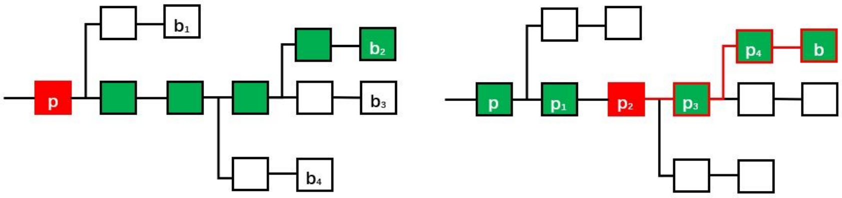
(1) Assume that the branch segment to be calculated is c→d. First, count Y_d; then, find the block that is voted by each miner for the last time in this range and evenly distribute the y stakes of each miner in Y_d to his each relevant transaction in the corresponding block.

(2) Count the y stakes assigned to the transaction whose b field corresponds to c→d interval; finally, sum up the results, which will indicate the weight of c→d branch.

If the branch length is longer than one voting cycle, it needs to be separated into small segments using one voting cycle as the unit and then calculated by segments. According to the aforementioned calculation method, the stakeholders represented by each miner can only vote once for any branch shorter than one voting cycle. Thus, if the branch obtains more than half of the total stakes of the whole system within a voting cycle, it is impossible to have a competitive branch. We denoted the branch as “Finalized,” and the block at the root of the branch as “save point.” All the blocks before the save point are irreplaceable, and all the blocks after the save point must be its descendants. The genesis block is the first save point, and the remaining save points are established based on the previous save point. The method is as follows:

(1) Assuming p is the latest save point and b_i is the newly generated block, the heaviest current branch according to the weight of $p \to b_i$ is determined as the main branch.

(2) Assuming the new block on the head of the main branch is b, p_j is the descendant and the length of $p_j \to b$ is shorter than one voting cycle. Calculate the weight of $p_j \to b$ successively; when it exceeds half of the total stakes, p_j becomes the new save point. Set $p = p_j$ and return to Step (1).



(1) (2)

Fig. 4 Process of determining the main branch (1) and save point generation (2)

3.5 Summary

According to the abovementioned consensus process, miners must earn more dispersed workers to acquire the randomly distributed signals of the voters in the network to obtain more advantages in the competition. Thus, competitive ability cannot be simulated on a single computer. This helps in avoiding competition by infinitely increasing the performance of mining machines and ensures fairness.

Similar to the PoS protocol, the proposed protocol does not require external resources. Unlike PoS protocol, the votes that take effect are already sealed in the existing blocks. This process resolves the double voting problem of the PoS protocol.

Moreover, stakeholders can hand over their work to miners via the voting process. Miners can help complete the competition even though low-stake users do not participate positively. Users with lower stakes only need to vote once in a voting cycle to ensure their stakes do not miss voting activities. This can essentially avoid the unequal wealth distribution issue. Users with higher stakes do not need to remain online, which also ensures high security. In other words, a cold wallet can be used for mining. Since the cost of stakeholders participating in network maintenance has been lowered, we can reduce the mining reward and avoid serious inflation.

Meanwhile, because the miner–worker form can be embedded in apps and websites, a large number of developers could be potential miners, which would reduce the threshold for starting mining, thereby improving blockchain sustainability.

4. Compete for Voters

Miners will attempt to win maximum voters to achieve fast block generation. Some miners may reach an agreement with some voters to mine cooperatively rather than using more workers as canvassers. In this case, the protocol provides alternatives for voters relative to miners in the transaction structure. Type A: accept vote canvassing from workers in the broadcast-request form; type B: designate one miner and make it win the vote; type C: designate one miner that belongs to the voter and work on mining individually. Regular miners will compete with miners having cooperating voters and the individual voters acting as miners. A and B miners compete to gather users, and type C and types A and B compete to gather stakes. Three types of miners essentially compete using the stakes gathered with three different abilities. Dynamic adjustment mechanisms are introduced via the reward distribution process to balance the mining power of these abilities. The risk that attackers will control the network by dominating one type of ability will be reduced effectively by this dynamic balance.

4.1 Wallet application

Wallet applications can guide or even control voters to choose a transaction type and voting targets due to a special user group. However, wallet applications would attempt to make voters vote for miners that belong to their suppliers, which would eliminate the chances for other ordinary miners to participate in the competition. Therefore, a mechanism to encourage wallet applications to select type A is required to provide a fair competition environment. In this case, the “wallet account” field is added to type A transactions and some profits are distributed to the suppliers of wallet applications; however, types B and C do not have such rewards. Any behavior that influences the fairness of the competition in type A should be considered malicious because wallet applications have legitimate income. All users and developers spontaneously play a supervisory role for such behaviors due to relevant interests. There is extra benefit if the wallet suppliers profit from the system, and this encourages developers to make better wallet applications, or, more importantly, it provides an incentive to build sidechain projects that may help build an extendable open system with multi-chains.

4.2 PoS variant

A variant can be derived for the PoS protocol when only type C miner is involved. Here, each miner mines using its own stake, which becomes pure PoS consensus. This variant system can also ensure fairness without introducing the common problems of the traditional PoS protocol, with the exception of the wealth distribution issue.

5 Reward distribution

Users may engage in different selections of mining types, which will affect the network structure if the distribution proportion of mining rewards is different. We can balance the distribution of users by dynamically adjusting the allocation proportion. The total rewards are divided into transaction fees and block rewards.

5.1 Transaction fees

Users need to pay a certain service fee for every transaction to compensate for the resources consumed when miners record and execute the transaction. To encourage more miners to use network dispersity for mining, the service fees in PoND are used as earnings to reward miners when the miners voted by nodes generate blocks successfully, unlike Bitcoin where the service fees for all transactions in the blocks generated are obtained directly by block generation miners. However, to encourage miners to pack transactions of higher service fees, we use a small portion of the service fee, e.g., 10%, as a reward to the miners who generated the current block.

When the service fee is considered as an income for distribution, miners and wallet are distributed at a fixed proportion, such as 15% each, and the remaining part is shared by all the voters participating in the block generation. Considering the case wherein the number of voters may be large when mining through network dispersity, we divide the earning into several parts and raffle among the voters several times. The proportion of stakes of the voters is the same as the probability of winning the lottery.

To ensure that sufficient service fees are considered as incentives, a minimum standard of service fee must be determined. However, the service fee is not charged compulsively; instead, a random operation based on constants with the result of 0/1 is conducted when the miner is packing a transaction with a fee lower than the minimum standard. This transaction can be included only when the result is 1. The probability that the operation result is 1 increases with an increase in the service fee of the transaction. It prompts users to pay adequate service fees and gives users the freedom to choose.

5.2 Block reward

To encourage more users with high stakes to participate in the maintenance, the system will issue another small amount of currency as the mining reward in addition to service fees. The amount of the mining reward affects the enthusiasm of stakeholders to participate in the type-C mining. Since type-C mining is an important method to adjust the balance, the value of the reward should be calculated dynamically based on the current participation rate and other parameters. For example, in the case wherein the miner and wallet accounts for 15% each and the allocation is balanced, the rewards of type-A and type-C mining are twice of the service fees.[7]

Mining reward value is an important balance parameter. It adjusts the number of participants in type-C mining activities and weakens the dominance of users' aggregation ability to prevent the centralization trend.

In addition, the following issues must be considered when setting the distribution ratio among various types of mining roles:

The following variables are set:

V: Voter earning; three types are V_a, V_b, and V_c, respectively.

M: Miner earning; only type A and type B have this item, which is M_a and $M_{b,}$ respectively.

W: Wallet earning; only type A has this item, namely W_a

R: Total earning of mining reward, i.e., the abovementioned mining reward value. Total earnings of three types are $R_a = V_a + M_a + W_a$, $R_b = V_b + M_b$, and $R_c = V_c$, respectively.

(1) To avoid the wallet application from getting involved in mining, the wallet earning must be larger than that obtained from its participation in mining, i.e., $W_a > M_b$.

(2) To prevent the cheating behavior that type-B miners and cooperating voters disguise together as type-A mining, the earnings of type-B voters must be larger than those of type-A voters, i.e., $V_b > V_a$.

(3) To prevent type-C mining from disguising itself as the other two types, the earnings of type-C mining should not be less than those of the other two, i.e., $R_c = \max\{R_a,R_b\}$. Since there is an additional W_a for type-A mining, the earnings are usually greater than those of type-B mining; therefore, it can be simplified as $R_c = R_a$, $R_a > R_b$.

To sum up, the earning allocation ratio is set based on following principles: $W_a > M_b$, $V_b > V_a$, $R_a > R_b$, and $R_c = R_a$.

In terms of V allocation, for type-A mining, it is the same as that of service fees, which is to raffle among voters according to the ratio of stakes; however, for type-B mining, V should be allocated by the miners.

5.3 Mining reward in side chain

In the open system structure of single main chain–multiple side chains, the additional issuance of currency can only be issued in the main chain. Therefore, the method to adopt the same reward rule in the side chain must be determined. The method of negative interest rate can be used to solve the problem and offset the additionally issued mining reward. In other words, the currency will be reduced in the same proportion for each account in the side chains with the passage of time to ensure that the total amount of currency in the side chains remains the same.

6 Frauds and attacks

(1) Filtered transaction. Miners only pack transactions that are beneficial.

Miners may want to gain more advantage by selecting transactions because the transactions included in each block could influence the competition environment.

First, transactions in each block only acquire a small percentage of the total amount; thus, minor changes in a single block may not have considerable influence on statistical results. To better address such problems, we must count the sum of the stakes of all voters in the given transaction, which will affect the next block generation interval. As the stake increases, the calculation period will be reduced, e.g., if the calculation period is 0.9–1.1 s, this would directly affect the generation speed of the next block. In this case, it would be better to pack maximum transactions, which could disincentive a miner's fraud. This parameter should be occasionally adjusted as per the average value .

(2) Simulate workers (i.e., sybil attack). Creating a large number of worker nodes via simulation and adding those virtual nodes to the P2P network will increase the success probability of canvassing.

To deal with this situation, we can control the process of creating P2P links, e.g., each node only needs to build connections with a certain number of nodes with the fastest response speeds.

(3) Super workers: If a worker is located near the backbone network, it will receive more votes because of the smaller network latency. If a large number of miners have opened up their supernodes, the miners who compete relying on the number of network terminals will lose competitiveness.

Although a variety of countermeasures can be added in the voting network, such as restricting the linking number of each IP address or adding the human-machine verification, the possibility will always exist due to interest motivations, which can also be seen as a threat to the fairness of network dispersity. Even so, this does not cause a great impact on the consensus mechanism itself as the network latency and bandwidth as an ability can also bring fair competition. If countermeasures are not taken, and in the end, if the miners want to win more votes, they will need to arrange more super worker nodes globally, which will not consume many social resources and will not violate the original intention of PoND design.

(4) 51% attack.

The system will be vulnerable to this attack if a user controls >50% of the stakes in the network, which is the same as the PoS protocol. However, the PoS system cannot maintain a high participation rate because users must run full nodes and keep them online to join the competition. Therefore, a lower online stake proportion reduces the level of security. In the PoND system, stakeholders can join network maintenance with only a single vote during a voting cycle. With a lower participation requirement, we can maintain a higher online proportion of the stake, which improves the system's security.

(5) Nothing at Stake (NaS) Problem

The problem caused by replacing high-cost computational resources with the stakes is the NaS problem. Users may compete simultaneously in multiple branches because of free participation in the competition, resulting in the reduced security of the main branch, i.e., double voting problems. A new branch can be built in the history block to replace the original branch, which is called the history attack. For example, both the long-range attack and costless simulation attack belong to the history attack.

For the problem of multiple voting, the votes in PoND have been stored in the block before counting. Choices made by the users cannot be changed, and there will not be a hidden competition branch; therefore, multiple voting is impossible.

PoND provides two levels of protection for the history attack. One is the save point. All history attacks that appear before the save point will be rejected. If the attack is launched after the latest save point, there is a second level of protection, i.e., branch priority determination strategy. The branch of PoND determines the priority based on the number of votes received. A branch with higher online stakes is certainly heavier; thus, the attack cannot succeed unless the historical stakes owned by the attacker exceed all the online stakes of the current main branch.

7 Conclusion

Compared to the PoW and PoS protocols, our model has the following benefits:

(1) No hashpower competition and no high-energy-consumption
(2) No such defects as multiple voting and history attack caused by the NaS problem
(3) Motivates sufficient competitive strength to maintain the security of network without falling into the problem of wealth centralization and inflation
(4) Possesses the finalization property and does not require any special node or extra expenses
(5) Provides incentive to develop extended projects, which ensures the sustainability and extendibility of the system
(6) The new mining method has abundant potential miner resources as the reserve

Conflict of Interest

Patent NO. : CN2018102289423.

Notes

[1] The theoretical winning probability is directly proportional to the number of online nodes.
[2] To improve the efficiency, we can add the transactions of a simple voting type; the users can also decide whether to participate in the voting in the current transaction within the voting cycle.
[3] Exclude the votes that chose the wrong branch. This ensures that the votes that chose the wrong branch do not benefit from the competition of generating blocks, prompting the miners to be more cautious in the y-vote.
[4] After a miner generates a block, the x stakes obtained will be emptied but the y stakes remain unchanged. The y stakes always maintain the counting interval of one voting cycle.
[5] Since this protocol may have a precise and objective counting on the activity stakes of the current branch, it is not necessary to rely on the previous speed of block generation to adjust the difficulty, and parameter d can be set directly according to the sum of online stakes.
[6] According to the GHOST protocol, the weight of a forked branch is equal to the weight of all sub-branches, but it can be simplified during the implementation using a method similar to Ethereum. Whenever there is an orphan block, the following blocks will reference the orphan block to increase the weight of branches so that the stakes that are already voted to this orphan block are not wasted. On the contrary, we do not need to reward the creator of orphan blocks since the orphan blocks themselves do not consume resources, and we do not need to reward their referrers since the reference will increase the weight of their own branches.
[7] Users will not be shared in revenues by miners and wallets when they are mining alone but will lose the opportunity to get abundant service fees. The higher the mining reward is, the more advantages there are for users to select the type-C mining, so the ratio of users selecting type-C mining can be adjusted by controlling the amount of mining rewards. Assuming that the miner and wallet share is 15% each, the service fee is f, and the mining reward is F, for a user to have the same earnings in type-A and type-C mining, it must meet $(f \times 90\% + F) \times (1 - 30\%) = F$; therefore, $F / f = 2.1$, which is approximately twice.

References

[1] Yj1190590 (/yj1190590). “PoND(Proof of Network Dispersity) BlockChain Project.” Github (accessed 29 April 2018) https://github.com/yj1190590/PoND/blob/master/README.md

[2] Paul Firth. “Proof that Proof of Stake is either extremely vulnerable or totally centralised.” BitcoinTalk.org (accessed 1 March 2016) https://bitcointalk.org/index.php?topic=1382241.0

[3] Vitalik Buterin. “Long-Range Attacks: The Serious Problem With Adaptive Proof of Work.” blog.ethereum.org (accessed 15 May 2014) https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/

[4] Yonatan Sompolinsky and Aviv Zohar. “Secure High-Rate Transaction Processing in Bitcoin” No Publisher (2013) https://eprint.iacr.org/2013/881.pdf

[5] Husam Ibrahim.“A Next-Generation Smart Contract and Decentralized Application Platform” Github (2018) https://github.com/ethereum/wiki/wiki/White-Paper#modified-ghost-implementation