Comparing Proof-of-Stake and Proof-of-Work Resource Consumption and Proof-of-Stake Improvements

Huirou Ma[1*]

[1*]2019081301011@std.uestc.edu.cn

[1]University of Electronic Science and Technology of China, Computer Science and Engineering (Cyberspace Security), 610000, Chengdu, Sichuan, China

Abstract—Consensus algorithms are getting more and more attention. It can help the majority of nodes in the blockchain to agree on the determination of new blocks. Proof-of-Work is a relatively mature consensus algorithm, which plays an important role in improving the security of Bitcoin. In addition, Proof-of-Stake is also a trending consensus algorithm. There have been many variants of Proof-of-Stake, and the direction of improvement still needs to be explored. Here, this paper reports the principle of the SHA256 one-way hash function often used in consensus algorithms, the core principle of Proof-of-Work, the core principle of Proof-of-Stake and its various variants such as Proof-of-Stake combined with Proof-of-Work and DPOS. At the same time, this paper simulates the mechanism of Proof-of-Work and Proof-of-Stake combined with Proof-of-Work to generate new blocks. The experiments in this paper show that Proof-of-Stake combined with Proof-of-Work reduces resource consumption compared to Proof-of-Work, and in Proof-of-Stake combined with Proof-of-Work, investing more coinages can help nodes get new blocks. The work in this paper can help provide a detailed elaboration and summary of the core technologies of Proof-of-Work and Proof-of-Stake and its variants. And show that it is effective that Proof-of-Stake combined with Proof-of-Work is effective in resource saving. It is hoped that this article will provide a more detailed reference for subsequent research on consensus algorithms.

Keywords-blockchain, Proof-of-Work, Proof-of-Stake, DPOS

1 INTRODUCTION

Nowadays, blockchain technology is getting more and more attention. Along with that, the consensus algorithm in it has also attracted much attention. In cryptocurrencies, consensus algorithms solve the problem of getting a majority of nodes to agree on a new block.

Before this paper, there have been many consensus algorithms that have proven to be effective. Among them, it includes the Proof-of-Work algorithm used in Bitcoin [1]. In addition, Proof-of-Stake to determine who is more likely to get a new block according to the invested coin value

ICIDC 2022, June 17-19, Qingdao, People's Republic of China

DOI 10.4108/eai.17-6-2022.2322749

has also been proposed, but this consensus algorithm is difficult to carry out on the actual blockchain due to many problems. There are many improved variants of Proof-of-Stake that can be applied in practice. For example, the consensus algorithm used in Peercoin [2] is a combination of Proof-of-Work and Proof-of-Stake. In addition, there are Delegated Proof-of-Stake [3] algorithms used in projects such as Bitshares and EOS [4].

However, at a time when Proof-of-Stake is attracting more and more attention and there are many variants, the material comparing Proof-of-Stake to Proof-of-Work and summarizing the available variants of Proof-of-Stake is still a little confusing. In some articles explaining the improvement of Proof-of-Stake, the basic technology, development history and improvement of Proof-of-Stake and Proof-of-Work are not clearly explained in detail. This is inconvenient for subsequent research.

Therefore, this paper does the following work: This paper summarizes the algorithm principle of the one-way hash function SHA256 [5], which is an important part of Proof-of-Work and Proof-of-Stake combined with Proof-of-Work. In addition, this paper also discusses the core technical principles of Proof-of-Work, Proof-of-Stake, Proof-of-Stake combined with Proof-of-Work and DPOS, showing that under different consensus algorithms, how nodes will agree on generating and obtaining new blocks. At the same time, this paper also simulates and compares the resource consumption between Proof-of-Work and Proof-of-Stake combined with Proof-of-Work through experiments. In addition, this paper also conducts experiments to show how long it takes for Proof-of-Stake combined with Proof-of-Work to obtain new blocks when the coinages invested is different, and then explains the impact of the invested coinages in Proof-of-Stake combined with Proof-of-Work. Besides, this paper also lists two tables to summarize and compare the characteristics of different consensus algorithms.

In this paper, the algorithm principle of the one-way hash function SHA256 is first described, and it is indicated that the Hash algorithms used after this paper are all SHA256. Then, this paper summarizes the core algorithm principle of workload proof. This consensus algorithm uses the one-way hash function SHA256 as the technical principle. When miners mine a new block, they must perform the operation of the SHA256 one-way hash function multiple times. Only the miners who find the solution first are rewarded, and the computing power of the remaining miners is wasted. This process also dynamically adjusts the block rate, making the blockchain secure and scalable. In addition, this paper also summarizes the core algorithm of traditional Proof-of-Stake and shows that in Proof-of-Stake, the probability of a node obtaining a new block is linked to the coin value invested. Not only that, this paper also summarizes the core algorithm of Proof-of-Stake combined with Proof-of-Work, which is similar to the workflow of Proof of Work, but the difficulty of a node to obtain a new block is linked to the basic difficulty and the invested coinages. This algorithm increases computing power consumption than traditional Proof-of-Stake, but also improves fairness. Additionally, this paper also shows the advantage of DPOS. This algorithm is more energy efficient and faster. At the same time, the results of the experiments in this paper can show that Proof-of-Stake combined with Proof-of-Work consumes less computing power than Proof-of-Work. In the case of Proof-of-Stake combined with Proof-of-Work, nodes will be more inclined to invest more coins rather than more computing power in order to compete for new blocks. This helps save computing power. It can be seen from the two tables listed that among the several consensus algorithms proposed in this paper, DPOS has the highest performance and efficiency, and POS does not need to consume resources.

This paper provides a detailed elaboration and summary of the core technologies of Proof-of-Work and Proof-of-Stake and their variants. At the same time, it also provides clear data support for the reduction in computing power consumption of Proof-of-Stake combined with Proof-of-Work compared to Proof-of-Work. This article can provide a more detailed reference for subsequent research.

2 Methods

2.1 SHA256

One-way hash functions compress messages or data into digests, reduce the amount of data, and fix the format of the data. The one-way hash function commonly used in consensus algorithms is often SHA256 [6]. For messages of any length, SHA256 produces a 256-bit hash value called a message digest. This digest is equivalent to an array of length 32 bytes, usually represented by a hexadecimal string of length 64, where 1 byte = 8 bits, and the length of a hexadecimal character is 4 bits.

In the process of constant initialization, the initial hash value $H_i^{(0)}$ (0<i<9) is taken from the fractional part of the square root of the first 8 prime numbers in the natural numbers, and the first 32 bit. The 64 hash constants are taken from the fractional part of the cube root of the first 64 prime numbers in the natural numbers, and the first 32 bits are taken.

The message needs to be preprocessed before computing the hash digest of the message. The process of complementing the message is to assume that the binary code length of the message M is L bits. First add a "1" at the end of the message. K is the smallest non-negative integer of the equation. Add K 0s. And need to keep adding 0s to pad to multiples of 512 bits. Operation process as in (1)

$$\mathrm{L} + 1 + \mathrm{K} \equiv 448 \bmod 512 \tag{1}$$

Divide the message into 512-bit segments and construct 64 words. The segment is subdivided into sixteen 32-bit word groups as big-endian integers W_0 ... W_{15}. The remaining words are obtained by the following iterative formula (2), (3), (4).

$$\sigma_0(x)=\mathrm{S}_7(x) \oplus \mathrm{S}_{18}(x) \oplus \mathrm{R}_3(x) \tag{2}$$

For this formula (2), specify the following symbols to describe: S_n means to rotate right by n bits, and R_n means to shift right by n bits.

$$\sigma_1(x)=\mathrm{S}_{17}(x) \oplus \mathrm{S}_{19}(x) \oplus \mathrm{R}_{10}(x) \tag{3}$$

For this formula (3), specify the following symbols to describe: S_n means to rotate right by n bits, and R_n means to shift right by n bits.

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} \quad (16 \le t \le 63) \tag{4}$$

For this formula (4), specify the following symbols to describe: W_t means the tth group as a big-endian integer. ($15 < t < 63$) σ_0 as in (2). σ_1 as in (3).

The following is the digest calculation main loop. First, variables a, b, c, d, e, f, g are initialized to $H_i^{(0)}$, where i = 0 ,..., 7. Then execute the compression function. The following logical functions are used: (5), (6), (7), (8).

$$\mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) \tag{5}$$

For this formula (5), specify the following symbols to describe: x, y, z make the input variables.

$$\mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \tag{6}$$

For this formula (6), specify the following symbols to describe: x, y, z make the input variables.

$$\Sigma_0(x)=\mathrm{S}_2(x) \oplus S_{13}(x) \oplus S_{22}(x) \tag{7}$$

For this formula (7), specify the following symbols to describe: S_n means to rotate right by n bits, and R_n means to shift right by n bits. x make the input variables.

$$\Sigma_1(x)=\mathrm{S}_6(x) \oplus S_{11}(x) \oplus S_{25}(x) \tag{8}$$

For this formula (8), specify the following symbols to describe: S_n means to rotate right by n bits, and R_n means to shift right by n bits. x makes the input variables.

$$\mathrm{T}_1 = \mathrm{h} + \textstyle\sum_1(e) + \mathrm{Ch(e,f,g)} + \mathrm{K}_t + \mathrm{W}_t \tag{9}$$

For this formula (9), specify the following symbols to describe: K_t is the tth key, t is 0 to 63. *Ch(x,y,z)* as in (5). $\sum_1$(x) as in (8). W_t as in (4).

$$\mathrm{T}_2 = \textstyle\sum_0(\mathrm{a}) + \mathrm{Maj(a,b,c)} \tag{10}$$

For this formula (10), specify the following symbols to describe: Maj*(x,y,z)* as in (6). $\sum_0$*(x)* as in (7).

Then assign values to the following variables: a=T_1+T_2, b=a, c=b, d=c, e=d+T_1, f=e, g=f, h=g. Calculate the intermediate result of the hash value of the jth group of the message, and assign the value according to the following equation (11):

$$H_0^{(j)} = a + H_0^{(j-1)}, H_1^{(j)} = b + H_1^{(j-1)}, H_2^{(j)} = c + H_2^{(j-1)},$$
$$H_3^{(j)} = d + H_3^{(j-1)}, H_4^{(j)} = e + H_4^{(j-1)}, H_5^{(j)} = f + H_5^{(j-1)},$$
$$H_0^{(j)} = g + H_6^{(j-1)}, H_7^{(j)} = h + H_7^{(j-1)} \tag{11}$$

After processing all groups, output a 256-bit hash value. The hash algorithms involved below are all SHA256.

2.2 The Proof-of-Work

The Proof-of-Work mining mechanism generates a new block by solving an arithmetic puzzle with an adjustable difficulty value [7]. In the Bitcoin blockchain, each block has a Merkle Tree, and the Merkle Root in the block header is generated by the hash values of all transactions in the block body. In Bitcoin's Proof-of-Work, first miners generate a transaction by themselves, and generate a Merkle root hash with all other transactions to be packaged through the Merkle tree algorithm. Then, the Merkle root hash and other components of the block header are assembled into the block header. Among the other components of the block header, there are nonce, hash value of the previous block, difficulty value, timestamp, version, etc. This assembled block header is then double-hashed to obtain the result. If the result is smaller than the preset difficulty value, it means the mining is successful [8].

Whether the result is smaller than the preset difficulty value is usually considered by the number of leading zeros. When a nonce is found that makes the leading zeros of the result match the difficulty value, it can be considered that it has solved the difficulty and a new block is generated. Under this consensus algorithm, the more computing power miners put in, the more likely they are to get new blocks. After getting a new block, miner will receive corresponding rewards. Other miners that did not find the nonce spent computing power without getting any reward. This is a huge waste of computing power and resources. The formula for judging to generate a new block is shown: (12)

$$\text{Hash } (Data, Nonce) < \text{ Difficulty} \tag{12}$$

For this formula (12), specify the following symbols to describe: Data means part of the data in the block header, such as hash value of the previous block, Merkle root hash, difficulty value, timestamp, version, etc. Difficulty means the difficulty value for generating the new block using Proof-of-Work. Nonce means a randomly selected value. Hash(x) means to hashing x using SHA256.

Also, the difficulty value of Proof-of-Work needs to be moderate. It can't be too hard or too easy. In the Bitcoin system, the difficulty is kept at an average of 10 minutes throughout the system to increase by one block. Bitcoin maintains such a block speed level by setting a difficulty target value in the block header. The formula for judging the difficulty to generate a new block in Bitcoin is shown: (13)

$$\text{Difficulty} = \text{Coefficient} * 2^{8 * \text{Index} - 3} \tag{13}$$

For this formula (13), specify the following symbols to describe: Difficulty means the difficulty value for generating the new block using Proof-of-Work. Coefficient means the last 6 digits of the Bitcoin Bits field. Index means the first two digits of the Bitcoin Bits field.

In Proof-of-Work mechanism, miners need to find a random number through continuous hash calculation, so as to satisfy the leading zero of the calculated hash value to satisfy its difficulty value. However, due to the increase in the computing power of miners today, in order to make the proof of work more effective, the difficulty value needs to be gradually increased. In Bitcoin system the difficulty value changes with the network, in order to ensure that a block can be generated every 10 minutes under different network environments. Shown as (14).

$$\mathrm{New} = \mathrm{Old} * (2016 * 10 \,/\, \mathrm{SumTime}) \tag{14}$$

For this formula (14), specify the following symbols to describe: New means the difficulty value of the new block. Old means the difficulty value of the previous block. SumTime means the real time it took to create the past 2016 blocks.

2.3 Original Proof-of-Stake

In the original Proof-of-Stake, people get new blocks based on the number of coin value they stake [9]. This means that the more coin value a person stakes, the more likely he is to get a new block. In the process of gaining new blocks, the Proof-of-Stake consensus algorithm will select based on the stakes invested by the investors. Selected investor will get a chance to get a new block. After getting a new block, investor will receive corresponding rewards. As shown (15).

$$\mathrm{P} = \mathrm{Submit}/\mathrm{Sum} \tag{15}$$

For this formula (15), specify the following symbols to describe: P means part of the data in the block, such as timestamp, transaction information, parent block hash, etc. Submit means the number of coin value invested by investors. Sum means the total amount of coin value in the network. (Sum > Submit)

2.4 Proof-of-Stake combined with Proof-of-Work

In order to improve the reliability of Proof-of-Stake, Proof-of-Stake combined with Proof-of-Work can be used [10]. The difference between Proof-of-Stake combined with proof-of-work and proof-of-work is that people with different coinages face different mathematical difficulties. In this consensus algorithm, a concept called coinage is also introduced. Coinages refers to the product of the value of coins and the duration of holding the coins. Coinages will increase over time. As shown (16).

$$\mathrm{CoinAge} = \mathrm{CoinTime} * \mathrm{CoinValue} \tag{16}$$

For this formula (16), specify the following symbols to describe: CoinTime means the time the coins were held. CoinValue means the number of coins. CoinAge means investors' stake in this consensus algorithm.

In this consensus algorithm, the mathematical problem to be solved has a basic difficulty. And everyone trying to get a new block faces a difficulty value that is adjusted based on the age of the coins they staked. If an investor invests a unit number of coinages, that is, 1 coin held for 1 second, then he will face a mathematical problem of basic difficulty. If an investor invests a lot of coinages, then he will face a math problem that is greatly reduced in difficulty. On this basis, investors who invest more coins will be more likely to get new blocks. At the same time, since coinages will continue to increase over time, the longer it takes for investors to mine new blocks, the easier it will be for new blocks to be obtained. After obtaining a new block, investors will receive the corresponding reward and the original invested coin value. However, the holding time of the invested coin value will be reset to zero. Investors who are not selected will withdraw the coin value they invested, and the holding time of the coin value will also be reset to zero. As shown (17).

$$\mathrm{Hash}\ (Data, Nonce)\ <\ \mathrm{Difficulty}\ * \mathrm{Coinage} \tag{17}$$

For this formula (17), specify the following symbols to describe: Nonce means a randomly selected value. Difficulty means the preset difficulty value. CoinAge means investors' stake in this consensus algorithm. Hash(x) means to hashing x using SHA256.

2.5 DPOS

Another improvement of Proof-of-Stake is the Delegated Proof-of-Stake consensus algorithm. Delegated Proof-of-Stake can be abbreviated as DPOS Unlike all POS nodes that can participate in bookkeeping, in the DPOS consensus algorithm, only designated nodes can participate in bookkeeping.[11] This will greatly increase the throughput of the system, as fewer nodes mean controllability of the network and nodes.

$$T\mathrm{BS}\ =\ \mathrm{Transactions/blockTime}\ *\ \mathrm{WitnessCount} \tag{18}$$

For this formula (18), specify the following symbols to describe: TBS means the number of transactions confirmed by the blockchain per second. Transactions means information about a transaction. It determined by the block size and physical bandwidth between the network state accounting nodes of the entire network. WitnessCount refers to the number of accounting nodes. WitnessCount determines the upper limit of physical bandwidth. Because the greater the number of accounting nodes, the higher the physical bandwidth requirements, and the higher the network stability requirements.

It can be seen from this formula that the number of nodes selected is inversely proportional to the number of transactions accepted by the blockchain per second. When the number of selected nodes is less, the TBS will increase, and the performance efficiency of the formula algorithm will be greater. Therefore, compared with the traditional Proof-of-Stake, this improved method of DPOS reduces the number of Witnesses, thereby improving performance efficiency. While DPOS is optimized in performance efficiency compared to Proof-of-Stake, it needs to prevent malicious nodes from being selected. DPOS is currently improving in the direction of reducing the probability of malicious nodes being selected [12].

3 Results and Discussion

In Proof-of-Work, hash the data and nonce in the block. When a nonce is found that makes the leading zeros of the result match the difficulty value, it can be judged that it has solved the difficulty and a new block is generated.

In Proof-of-Stake, it is just weighted according to the coin value invested by different investors, representing their respective probability of getting a new block. In Proof-of-Stake combined with Proof-of-Work, also hashed the data and nonce. When a nonce is found that can make its result smaller than the difficulty adjusted by coinage, it is judged that a new block is generated. The difficulty here is negatively correlated with the coinage. The greater coinages invested, the less difficulty investors face.

3.1 Different consensus algorithms to generate a new block

In Proof-of-Work, hash the data and nonce in the block. When a nonce is found that makes the leading zeros of the result match the difficulty value, it can be judged that it has solved the difficulty and a new block is generated. In Proof-of-Stake, it is just weighted according to the coin value invested by different investors, representing their respective probability of getting a new block. Investors in this group are randomly selected according to their weights, and the selected investors will get the new block. In Proof-of-Stake combined with Proof-of-Work, also hashed the data and nonce. When a nonce is found that can make its result smaller than the difficulty adjusted by coinage, it is judged that a new block is generated. The difficulty here is negatively correlated with the coinage. The more coinages invested, the less difficult it is for investors.

In the experiment of this paper, the process of node computing and generating new blocks in Proof-of-Work and Proof-of-Stake combined with Proof-of-Work is simulated by writing programs. When a node generates a block first, it means that it has obtained the block. The number of hash computations is used to measure the computational resources consumed in the process. The more hash calculations are performed, the more resources are consumed.

In this paper, multiple experiments are performed by changing the data and keeping other settings unchanged. At the same time, the outputs were grouped according to each experiment. In this paper, the results of seven groups were selected for analysis.

As shown in Fig. 1, in the seven experimental groups, the results are that the hash calculation times of Proof-of-Stake combined with Proof-of-Work are far less than the hash calculation times of Proof-of-Work. The resources consumption at which Proof-of-Stake combined with Proof-of-Work generates new blocks is much greater than that for Proof-of-Work generates new blocks, which reflects the significant improvement in resources saving of Proof-of-Stake combined with Proof-of-Work relative to Proof-of-Work.

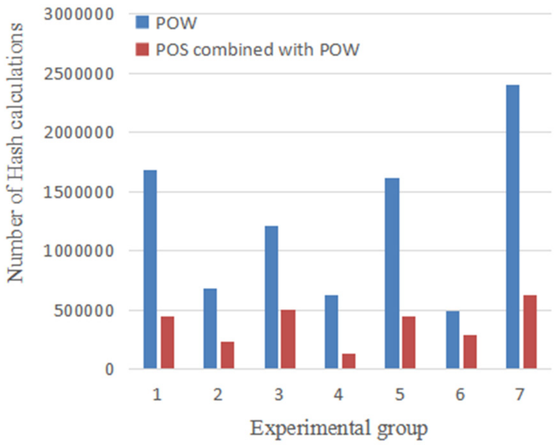

Figure 1. Number of Hash calculations for different consensus algorithms to generate a new block.

3.2 Different invested coinages to generate a new block in Proof-of-Stake combined with Proof-of-Work

In Proof-of-Stake combined with Proof-of-Work, the coinages invested is determined by the value of coins invested and how long the coins are held. In this experiment, the mechanism of Proof-of-Stake combined with Proof-of-Work to generate new blocks is simulated. The basic difficulty setting and other conditions remain unchanged, and the input coinages is changed to simulate the process. During the experiment, the invested coinages was set to 1000 and 1100 respectively.

As shown in Fig. 2, compared with nodes with input 1000 coinages, nodes with input 1100 coinages have less time to generate new blocks. The node with 1000 coinages takes 98s to obtain a new block, while the node with 1100 coinages takes only 13s to obtain a new block. The more coinages investing, the less time needed to generate a new block. And while the time consumption is greatly reduced, the consumption of computing resources is also reduced. It can be seen that in reality, people who invest more coinages, will more likely get a new block. In order to get new blocks, miners will choose to invest more coinages instead of investing more computing power. Compared with Proof-of-Work, it no longer needs to increase the speed of opening new blocks by increasing the number of miners and power consumption, which decreases the electric consumption.

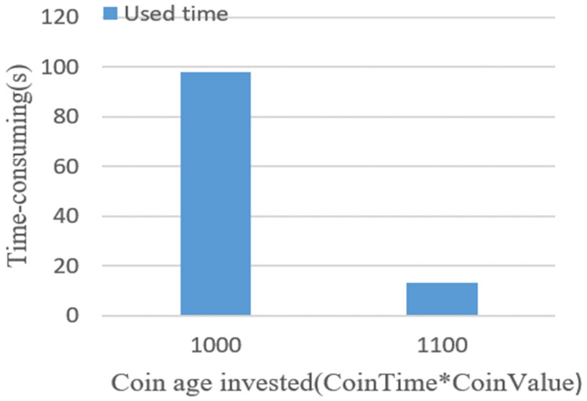

Figure 2. Time consuming for different coinages to generate a new block in Proof-of-Stake combined with Proof-of-Work

3.3 Comparison and summary of Proof-of-Work, Proof-of-Stake and Proof-of-Stake combined with Proof-of-Work

In addition to the above two experiments, this paper also summarizes and compares the points that need to be compared for the three consensus algorithms Proof-of-Work, Proof-of-Stake and Proof-of-Stake combined with Proof-of-Work [13].

As shown in Table 1, Proof-of-Work is abbreviated as POW. Proof-of-Stake is abbreviated as POS. Proof-of-Stake combined with Proof-of-Work is abbreviated as POS Combined with POW.

Proof-of-Stake requires almost no resource consumption, while Proof-of-Work and Proof-of-Stake combined with Proof-of-Work are still required. Proof-of-Stake creates new blocks faster than Proof-of-Work and Proof-of-Stake combined with Proof-of-Work [14].

TABLE 1. COMPARISON OF PROOF-OF-WORK, PROOF-OF-STAKE AND PROOF-OF-STAKE COMBINED WITH PROOF-OF-WORK

Point of Comparison	**Consensus Mechanism**		
	POW	***POS***	***POS Combined with POW***
Resource consumption	Yes	No	Yes
Speed of generating blocks	Slow	Fast	Slow

3.4 Comparison summary of Delegated Proof-of-Stake and Proof-of-Stake

Delegated Proof-of-Stake is a variant of Proof-of-Stake. This paper concludes the optimization of Delegated Proof-of-Stake by summarizing and sorting out.

As shown in TABLE 2, Proof-of-Stake is abbreviated as POS. Delegated Proof-of-Stake is abbreviated as DPOS.

Proof-of-Stake requires almost no resource consumption, while Proof-of-Work and Proof-of-Stake combined with Proof-of-Work are still required. Proof-of-Stake creates new blocks faster than Proof-of-Work and Proof-of-Stake combined with Proof-of-Work.

Both consensus algorithms in TABLE 2 are well decentralized. The performance efficiency of Proof-of-Stake is lower than that of Delegated Proof-of-Stake, and fork problems may occur [15]. Delegated Proof-of-Stake is better than Proof-of-Stake in terms of performance efficiency and fork.

TABLE 2. COMPARISON OF PROOF-OF-STAKE AND DELEGATED PROOF-OF-STAKE

Consensus Mechanism	**Attributes**		
	Performance Efficiency	***Fork***	***Decentralization***
POS	Low	Possible	Yes
DPOS	High	Difficult	Yes

4 CONCLUSION

This paper summarizes and analyzes the algorithm principle of the one-way hash function SHA256, and the core technical ideas of Proof-of-Work, Proof-of-Stake, Proof-of-Stake combined with Proof-of-Work and DPOS, and also shows Proof -of-Stake combined with Proof-of-Work compared to the respective computing power consumption of Proof-of-Work, and the impact of coinage investment in Proof-of-Stake combined with Proof-of-Work.

In this paper, the algorithm principle of the one-way hash function SHA256 is shown. Then, this paper summarizes the core algorithm principles of Proof-of-Work. This consensus algorithm uses the one-way hash function SHA256 as the technical principle. When miners mine a new block, they must perform multiple operations of the SHA256 one-way hash function. A random hash value in a block starts with one or more 0s. As the number of 0s rises, the amount of work required to find this solution grows exponentially, and miners try to find this solution over and over again. This process also dynamically adjusts the block generation speed. Proof-of-Work consumes and wastes a lot of computing power. Additionally, this paper also summarizes the core algorithm of traditional Proof-of-Stake, showing that in traditional Proof-of-Stake, the probability of a node obtaining a new block is directly linked to the coin value invested, but this will cause the rich to become richer. Besides, this paper also summarizes the core algorithm of Proof-of-Stake combined with Proof-of-Work. The difficulty of a node obtaining a new block is linked to the basic difficulty set by the system and the age of the coins invested. The easier it is to get new blocks. Compared with Proof-of-Work, Proof-of-Stake combined with Proof-of-Work greatly reduces computing power consumption, solves a major problem faced by Proof-of-Work, and saves resources. In addition, this paper also explains that DPOS reduces the network delay by reducing the number of participating nodes, and further improves the performance efficiency of the blockchain.

This paper clarifies and summarizes the technical ideas of Proof-of-Work and Proof-of-Stake and its variants. It also compares the computational power consumption of Proof-of-Work and Proof-of-Stake combined with Proof-of-Work. It shows that Proof-of-Stake combined with Proof-of-

Work has a great improvement in computing power saving. This article can provide a more detailed reference for subsequent research and improvement of blockchain technology and consensus mechanisms such as Proof-of-Work and Proof-of-Stake. Proof-of-Stake can continue to improve in future research. A more efficient, safer and more economical consensus algorithm can be realized through the combination of various consensus algorithms.

REFERENCES

[1] S. NAKAMOTO,“Bitcoin: A peer-to-peer electronic system,” https://bitcoin.org/bitcoin.pdf, 2008.

[2] S M. KING S, NADAL ,“PPcoin:Peer-to-peer crypto-currency with Proof-of-Stake”, https://www.semanticscholar.org/paper/PPCoin%3A-Peer-to-Peer-Crypto-Currency-with-KingNadal/0db38d32069f3341d34c35085dc009a85ba13c13, August 19, 2012.

[3] J. WEN Xiao-lin, LI Chang-lin, ZHANG Xin-yi, LIU Shang-song, ZHU Min, “Visual Analysis Method of Blockchain Community Evolution Based on DPoS Consensus Mechanism,” Computer Science, vol. 49, pp. 328-335, 2022.

[4] GRIGGI. EOS—An introduction https://eos.io/documents/EOS_An_Introduction.pdf.,2017.

[5] Dmitry Khovratovich, Christian Rechberger & Alexandra Savelieva. Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family (PDF). IACR Cryptology ePrint Archive. 2011.

[6] J. HE Runmin, “Research and Improvement of One-way Hash Function SHA-256,” Information Technology, vol. 37, pp. 22-25, 2013.

[7] J. Lepore Cristian,Ceria Michela,Visconti Andrea,Rao Udai Pratap,Shah Kaushal Arvindbhai,Zanolini Luca, “A Survey on Blockchain Consensus with a Performance Comparison of PoW, PoS and Pure PoS”, Mathematics, August,2020.

[8] J. HAN Xuan, LIU Yamin. “Research on Consensus Mechanism of Blockchain Technology,” Netinfo Security, pp:147-152, 2017.

[9] J. Cong T. Nguyen, Dinh Thai Hoang, Diep N. Nguyen, Dusit Niyato, Huynh Tuong Nguyen, Eryk Dutkiewicz, “Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities,” IEEE Access, vol. 7, pp. 85727-85745, 2019.

[10] J. WU Mengyu , ZHU Guosheng , WU Shanchao, “Improved consensus mechanism of blockchain based on proof-of-work and proof-of-stake”, Journal of Computer Applications, vol. 40, pp. 2274-2278, 2020.

[11] J. Hu Qian, Yan Biwei, Han Yubing, Yu Jiguo, “An Improved Delegated Proof of Stake Consensus Algorithm,” Procedia Computer Science, vol. 187, pp. 341-346, 2021.

[12] D. MA Chaoyu, “Research on Improved Consensus Mechanism of Delegated Proof of Stake in Blockchain”, 2020.

[13] D. ZHA Peng, “Research and Implementation of Blockchain Proof of Work Consensus Algorithm”, 2021.

[14] J. YU Benguo, GONG Shiming, PANG Xiaoqiong, NIE Mengfei, CHEN Wenjun, YANG Ting, “Fair and stable minimum proof consensus mechanism”, Computer Engineering and Applications, vol. 56, pp. 63-68, 2020.

[15] J. LIU Tongtong, “Research and analysis on consensus mechanism of blockchain,” Information and Communications Technology and Policy, pp. 26-33, 2018.