Fair consensus algorithm: Deb consensus algorithm

(A fair consensus algorithm: deb consensus algorithm)

Contents

1. Overview

In 2008, Satoshi Nakamoto developed a decentralized(decentralized) P2P crypto currency system, Bitcoin, using the distributed ledger concept and consensus algorithm, Proof of Work.) After, 2014 BUTALIN overcame the limitations of Bitcoin "Global Trust Computer (A trust world computer)"In Ethereum developed.

The most important core technology of blockchain is consensus algorithms among nodes that do not trust each other. Both Bitcoin and Ethereum use proof-of-work methods as consensus algorithms. However, for consensus algorithms that use proof-of-work methods, the computing power held by the node (computing power) Mined by(mining)Odds are determined. This has the disadvantage of weakening the decentralized nature that blockchain seeks., The problem of centralization of Bitcoin is emerging as a reality. With this reason, Ethereum has now demonstrated the consensus algorithm in the proof of work (PoS: Proof of State) The situation that is switching to a way. However, even in the case of equity proof, the decentralization characteristics of the nodes' holdings are fundamental to the sustainable nature of the shares. The debate about how to prove equity will fundamentally lead to capitalist problems is also the reason for this.

We first want to define and analyze the decentralized nature of consensus algorithms, the core source technology of blockchain, into a fairness concept. Simply put, the fairness of the consensus algorithm is the condition of the nodes to be mined. (Computing power, holding stakes, etc.), meaning proportionality of mining probabilities.

We will propose a deb consensus algorithm that maximizes fairness while being a public blockchain (AndUS) that maintains sustainable decentralization characteristics.

The AndUS blockchain is basically based on Ethereum. In other words, AndUS Blockchain is a representative public blockchain (public blockchain) While maintaining the structure of the Ethereum, it is a public blockchain that maintains sustainable decentralization and greatly improves speed. To date, public and private or consortium blockchains (private or consortium blockchain) many blockchains are being proposed. It is thought that the ethereum blockchain satisfies the philosophical characteristics of the original blockchain. In particular, Decentralization, which is the primary purpose of Ethereum P2P Business ecosystem(ecosystem)the most important and essential philosophy of blockchain.

On the one hand, it is said that the connection between mining and crypto currency issuance is different from the deb consensus algorithm and the proof of work and proof of equity used by the existing public blockchain. Existing consensus algorithms give crypto currency issuance authority as a reward to successful nodes in order to maintain their mining participation.

However, in the case of deb consensus algorithms, it has nothing to do with mining and issuing crypto currencies. In other words, it is a consensus algorithm for the development of the first public blockchain that mining and crypto currency issuance are irrelevant. miners are not given the right to issue crypto currencies. It is a method of compensating for a portion of the participation fee and transaction fees of the paid mining league, which will consist of nodes that wish to participate in mining. The essential reason for this configuration is also linked to making the nodes independent of the mining conditions for sustainable decentralization. In other words, in order to ensure the fairness of the mining operations, the mining process is very low cost to ensure that all nodes are fair, so that a high-value compensation system is not required.

In particular, deb consensus algorithms also have the advantage of not generating forks, unlike traditional public blockchains. This means that block creation is the finality of the block. (finality)to ensure that.

2. Fairness of consensus algorithms

The purpose of the deb consensus algorithm is to maintain a sustainable decentralization characteristic by ensuring fair mining probabilities for all nodes, regardless of the conditions of the nodes they want to mine. (computing power, holding stakes, and etc.)

To do this, we first define the fairness of consensus algorithms and analyze the fairness of existing public blockchains.

Definition: Fairness of consensus algorithms

The fairness of consensus algorithms is defined by correlation between the probability of mining nodes and the conditions they have (computing power, equity, etc.).

For example, in the case of proof-of-work methods used in Bitcoin and Ethereum, the probability of mining nodes is determined for the computing power held by the node. In other words,

$$\text{Mining Success Probability of Node} = \frac{Owned\ computing\ power}{Sum\ of\ computing\ power\ held\ by\ all\ nodes}$$

In the case of Bitcoin, which adopts a proof-of-work method, the probability of a typical node mining is almost zero, depending on the birth of the mining plant and group. This has led to controversy that Bitcoin is being centralized.

And in the case of the proof of equity method to be used in Ethereum, the probability of mining nodes is determined by the stake held by the node. In other words,

$$\text{Mining Success Probability of Node} = \frac{Owned\ stake}{Total\ amount\ of\ cryptocurrency}$$

In the case of the equity proof method is a situation in which the problem is pointed out that the logic of the typical capital is applied to the probability of mining according to the holding stake is determined.

3. deb consensus algorithm

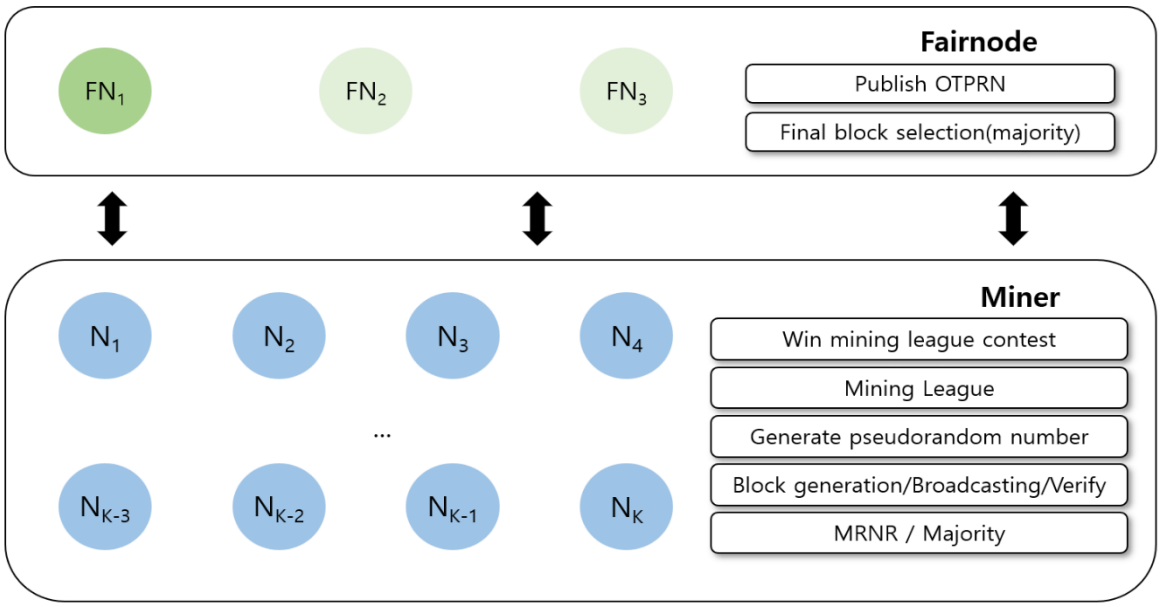
In the case of the existing proof of work and proof of equity consensus algorithm, the probability of mining of the mining node is proportional depending on the computing power and the shares held by the mining node, which tells us that it is not fairness to miners who want to participate in the blockchain from a mining perspective.

The deb consensus algorithm is a consensus algorithm that addresses these unfair problems and ensures fair mining opportunities. First, to ensure fair mining opportunities, all nodes that want to be mined (Computing power, holding stakes, etc.) fair mining opportunities regardless of.

To this end, the deb consensus algorithm introduces the concept of fairnode, as opposed to proof of work and proof of equity. Of course, P2P Based on deb, it does not assume the reliability of a fair node to maintain the characteristics of the consensus algorithm. In other words, Fair node 3 Trust organizations of (TTP: Trusted Third Party) not, It's just P2P Think of it as a simple, special node that works with the nodes in the network to support consensus algorithms. The role and safety of fair nodes will be discussed at a later date.

The deb consensus algorithm works with three basic principles: paid mining leagues, maximum random number rules (MRNR: Maximum Random Number Rule, largest random number) and majority resolution principles. A paid mining rig is a specific number of nodes that want to be mined. (Yes,100 people) a group of mining nodes consisting of nodes. Of course, nodes who want to participate in the mining league sit in the mining league, which is realistically small enough to participate in the mining league. (Yes,100 yen) must pay the participation fee. And in a group of nodes participating in a paid mining rig, the rule in which each node generates a block is the maximum random number rule. And how to determine the final miner, in other words, the method of determining the final block consists of a majority rule through cooperation between the fair node and the nodes participating in the mining league.

The overall configuration of the deb consensus algorithm is as follows:



< Figure 1 > Full configuration concept

For fair nodes, it can be configured as one or more nodes.

3.1 deb consensus algorithm full process

The entire process of deb consensus algorithm consists of three phases: paid mining rig configuration, block creation(mining), and final block consensus.

☐ **Organized a paid mining league**

① The node that wants to mine provides its own access information to a fair node.

② Fair nodes are distributed to all $OTPRN$ nodes for mining rig configuration.

③ Nodes who wish to participate in the mining league shall $OTPRN$ determine whether they are eligible to participate in the mining league by referring to the distribution of the fair node.

④ Mining nodes selected as participants of the Mining $OTPRN$ $JoinTx$ League will be created, including for the construction of mining leagues.

⑤ Broadcast to all $JoinTx$ nodes.

⑥ Refer to only mining nodes JoinTx selected as participants of the Mining League.

☐ **Block creation (mining)**

The mining node participating in the mining league creates $difficulty$ the basis for the final block selection.

$$\text{difficulty} = \{0 \leq \text{n} \leq \text{JoinNonce} | \text{MAX}(\text{CSPRNG}(\text{n}, \text{OTPRN.rand}, \text{coinbase}, \text{P\_blockHash}))\}$$

※ If you have applied for a mining rig to equalize the probability of mining, but have not been selected as a miner, you will generate $difficulty$ multiple numbers if you are not selected.

The mining node generates a block, $difficulty$ including in the block header.

Broadcast blocks created for all nodes.

☐ **Consensus algorithm**

The basic principle of block consensus is the final block consensus process by majority resolution, in which the largest number(MRNR: Maximum Random Number Rule, the largest random number) rule and the node and the fair node work together.

① The node selects and signs the largest block of the block $difficulty$ that it receives and sends it to the fair node.

② The fair node decides and signs the most selected blocks of the blocked sent in accordance with the majority resolution principle and sends them to the nodes.

③ The mining node broadcasts to the entire node after verifying that the block received from the fair node is a block selected by a large number.

④ Each node recognizes a fair node and a block signed by a majority as the final block and adds it to the blockchain.

3.2 Paid Mining League Configuration Detailed Process

For safety and efficiency, the method of organizing a paid mining rig is conducted by adjusting the number of fair nodes and nodes themselves and applying for participation in the mining league.

☐ **Selected as a participant in a paid mining league**

① Nodes that want to be mined provide node information to a fairnode.

Table 1. enodeCoinbase structure

Field name	Description
$enode$	Enode value of mining node
$coinbase$	The address of the account to participate in mining.
$port$	Port number to communicate with fair node

② A fair node distributes a $transOTPRN$ structure to all nodes, depending on the block creation cycle.

Table 2. transOTPRN structure

Field name	Description
$OTPRN$	The structure that mining nodes refer to when mining
Sig	$OTPRN$ The signing of a fair node for

Table 3. OTPRN structure

Field name	Description
num	OTPRN issue number
rand	One-time pseudo-random numbers that fair nodes periodically deploy
CMiner	The number of nodes that are maintaining a fair connection to the mine to perform mining.
Timestamp	Local time for fair nodes

③ Mining candidate nodes refer to the structure deployed by the fair node $OTPRN$ to determine if they can participate.

A. Set the system setting variable Mminers to the number of participants in the maximum mining

B. The mining node sets $OTPRN$ the Cminers as a avoiding water, indicating the total

number of mined nodes that the fair node propagated, which indicates the intention of mining.

C. Set the amount obtained by computing two values to Div
$Div = CMiner \div MMiner$

D. $enodeCoinbase.\text{coinbase}$ Using the sum of the $OTPRN.\text{rand}$ XOR operations as the seed of the random function to derive random values* is a random function.

$$\text{rand} = \text{RAND}(\sum_{i=0}^{19} enodeCoinbase.coinbase[i] \oplus OTPRN.rand[i]))$$

※ RAND is a random function

E. $rand$ modular operations to determine that mining participation is possible when the following conditions are met. div
rand % div == 0

☐ **Organized a paid mining league**

① Mining nodes that can participate in the mining league create $OTPRN$ a structure, including the structure, $JoinTx$ and broadcast it to all nodes.
* $JoinTx$: Application transactions for participation in mining leagues that result when a node wants to participate in a mining league

Table 4. JoinTx structure

Field name	Description
Tx	Same as ethereum transaction structure except for the field below to : $Fiarnode's\ address$ $data$: $JoinTxData$

Table 5. JoinTxData Structure

Field name	Description
$JoinNonce$	Add 1 to the account $JoinNonce$
$OtprnHash$	Hash value received from a fair node $OTPRN$
$FairnodeSig$	$transOTPRN.sig$

$Timestamp$	Local time for mining nodes
$NextBlockNum$	Number of blocks to be mined

② Collect only mining nodes that $JoinTx$ have participated in the Mining League.

③ The mining node organizes its own $JoinTx$ mining rig by listing the collected ones.

3.3 Block creation process: mining process

Use fair nodes and pseudo-ins for fair and efficient $Difficulty$ mining.

□ Difficulty generation

① The mining node generates $difficulty$ using a reference to the OTPRN structure received from a fairnode in the participant selection process.

$$\text{difficulty} = \{0 \le \text{n} \le \text{JoinNonce} | \text{MAX}(\text{CSPRNG}(\text{n}, \text{OTPRN.rand}, \text{coinbase}, \text{P}_{\text{blockHash}}))\}$$

※ $JoinNonce$: $JoinNonce$ for other purposes of, As the mining node participates in the mining rig, it performs the function of increasing the probability of mining. To achieve this goal, $JoinNonce$ As many as different $difficulty$ can be generated and the largest value of which can be used to create blocks.

※ $OTPRN.\text{rand}$: A one-time pseudo-random number deployed by a fair node prevents mining nodes from generating any value that is favorable for mining $difficulty$

※ $coinbase$: To have the mining nodes create differently by the address used to mine $difficulty$

※ P_BlockHash : Hash value of previous block (i) To prepare for the distribution of a fair node favorable to a particular mining node, and $OTPRN.\text{rand}$ (ii) to allow the mining node to create one that is dependent on a particular block. $difficulty$

② The mining node creates a $difficulty$ transaction by selecting the $difficulty$ largest of its own creations.

$$\text{difficulty} = \text{MAX}(0 \le \text{n} \le JoinNonce | \{difficulty_n\})$$

□ Block creation and broadcasting

① The mining node generates random numbers by referencing their block headers $OTPRN.\text{rand}$, $JoinNonce$ and other data, and then records the random number and its own in the block header. Exist within a block and is recorded by a fair node in the future FiarNode.Sig and Voters.

Table 6. Block structure

Separated	Field name	Description
$Header$	$UncleHash$	Remove
	$JoinTxHash$	$JoinTx$ List hash value
	$GenTxHash$	Tx List hash value
	$JoinReceiptHash$	$JoinTx$ Receipt hash value of
	$GenReceiptHash$	Tx Receipt hash value of
	$Difficulty$	Random value generated by utilizing the mined node received from a fair node $OTPRN.\mathrm{rand}$
	$nonce$	$JoinNonce$ The value of the mining node
	$FairNodeSig$	A fair node is a value signed by a number of mining nodes, including selected blocks and proof data.
	$VoterHash$	$Voters$ Hash value
$Body$	$JoinTxs$	Tx List
	$GenTxs$	$JoinTx$ List
	$Voters$	Address and signature of nodes that voted on the block (multiple)

Table 7. Voters structure

Field name	Description
$addr$	The address of the mining node that voted for the block.
sig	Signature of mining node
$difficulty$	$difficulty$ Used to verify that the value created by the mining node is correct, selected by a $difficulty$ future majority.

② The various transactions collected by the mining node are included in the block, and then the block is created.

③ The mining node broadcasts the generated blocks to other mining nodes.

3.4 Consensus algorithm

Block consensus is basically $MRNR$ based on the principle of majority resolution. In other words, originally, mining nodes broadcast their own created blocks. Select the largest designated block of the block received to you $difficulty$(MRNR) and then sign and send to a fair node.

A fair node selects a block selected by a large number of the blocks it receives (majority resolution principles) and includes the address and signature of the mining nodes within the block and then signs itself. And if a fair node propagates the block to the mining node, the mining nodes will have the block in accordance with the principle of majority resolution., Verifies whether a fair node has been signed, and then add it to the ledger. As a result, the block is determined as the final block., Mining nodes propagate the block to the network.

☐ Validation phase

① Make sure that the $OTPRN$ propagation cycle and the block creation cycle match.

② Verify the $OTPRN$ integrity and signature of fair nodes.

③ Make sure that the mined node is eligible to participate in the mining league.

④ Make sure that the mining node can pay for the mining league.

⑤ make sure that $difficulty$ is created correctly.

☐ Block agreement

① The mining node selects the block(MRNR) that is the largest $difficulty$, signs it, and sends it to a fair node.

② A fairnode is selected and signed as the final block and then sent to the nodes in accordance with the principle of majority resolution of the sent block. For future verification, a fair node will include and sign the addresses and signatures of the mining nodes that voted on the block.

③ The fairnode broadcasts the block to the mining node. Mining nodes are sure that the

received blocks meet the principle of majority resolution., Verify that a fair node's signature is included, then add it to your ledger and broadcast the block.

④ Similarly, the general nodes that received the above block nodes (that did not participate in mining) also go through the same verification procedures as the mining nodes performed, and then add the block to the ledger.

⑤ Miners are provided with incentives as follows:
Incentives= Transaction fees + Total participation fee for mining league participants

⑥ Adjust the mining probability of mining league participants.
- Mining Success Node : $\mathrm{Tx.Join\_Nonce} = 0$
- Mining Failure Node : $\mathrm{Tx.Join_{Nonce}} = \mathrm{Tx.Join_{Nonce}} + 1$

4. The role and reliability of fair nodes

Consensus algorithms in Bitcoin and Ethereum do not use fair node concepts. However, deb In the case of consensus algorithms, the concept of fair nodes was introduced to maintain sustainable decentralization characteristics. Of course, deb consensus algorithm does not assume the reliability of a fair node in order to operate.

The Fair Node is only responsible for the efficiency, block consensus, and finality cooperation of the paid mining league configuration.

☐ **The role of fair nodes**

① Random selection of paid mining league participants

② Final block consensus cooperation through mutual checks with nodes

Most importantly, the deb consensus algorithm does not depend on the reliability of a fair node. Through mutual checks between fair nodes and blockchain nodes, the safety of the blockchain can be secured without ensuring the reliability of fair nodes.

The fairness of the deb consensus algorithm using a fair node can be thought of as follows:

$$\text{Mining Probability of Node} = \frac{1}{Total\ number\ of\ nodes\ desired\ to\ be\ mined}$$

5. Performance

The performance of the deb consensus algorithm can be dynamically determined by the number of paid mining league configurations and block creation cycles.

For example, if you have 100 mining leagues, the expected performance is as follows:

Table 8. deb Consensus Algorithm's Performance

	deb consensus algorithm
Block Size	4.5MB ~ 9MB
TPS	1000 TPS
Creation cycle	30sec ~ 1min

In particular, reducing the network connection load between fair nodes and paid mining league nodes to reduce block creation time can further reduce the block creation time. For example, the number of blocks generated by a paid mining rig that is configured once 10 block creation time 10 will be shortened in seconds.

6. Deb consensus algorithm features

The purpose of the deb consensus algorithm is to solve the problem of centralization of blockchain consensus algorithms that can be caused by the unfairness of the current consensus algorithm. In other words, Proof of work, an existing consensus algorithm is to prove your stake, and the biggest difference between the consensus algorithms is that sustainable decentralization can be maintained. This means that it is a fair consensus algorithm that does not depend on the conditions of the nodes that want to be mined.

In addition, in the case of the consensus algorithm of the existing public blockchain, but the mining and crypto currency issuance to generate the block is linked, in the case of deb consensus algorithm, mining and crypto currency issuance is irrelevant. In other words, this means that the amount of crypto currency issued earlier is the total amount of currency.

This is the first consensus algorithm that allows the public blockchain to be configured while monopolizing the right to issue crypto currencies.

On the other hand, the advantage of deb consensus algorithm has the advantage that the fork does not occur if the finality is one block.

□ **Deb consensus algorithm features**

① Maintaining sustainable decentralized characteristics(fairness)

② No mining and cryptocurrency issuance (cryptocurrency issuance can be exclusively available)

③ Finality guarantee of one block without fork

④ High-speed performance over 1,000 TPS

7. Conclusion

The deb consensus algorithm is the first consensus algorithm that solves the centralization problem due to the characteristics of the existing consensus algorithm, proof of work and proof of equity (conditions of nodes). This is a key concept that can achieve sustainable decentralization, which is the original purpose of blockchain.

Itis also the first public blockchain that does not link with the issuance of new crypto currencies as a compensation system for the operation of the public blockchain. This is linked to sustainable decentralization characteristics by minimizing mining costs for miners.

In particular, the deb consensus algorithm is a key source technology that enables the most performing public blockchain and AndUs blockchain to be realized while maintaining sustainable decentralization of the proposed public blockchain to date.

Table 9. Key Public Blockchain Performance Comparisons

	Bitcoin	Ethereum	AndUs Blockchain
Consensus algorithm	Proof of work	Proof of work	Deb consensus algorithm
TPS	7	12~15	More than 1,000
Finality	10 minutes	About 3 minutes	10 seconds to 1 minute

We will maintain a sustainable decentralization that will have the same mining probabilities under fair conditions.

History of changes

2019.05.31. - Document Version 0.91

- The cause of the change: Mining participation transactions are not included in the block due to low priority to include blocks compared to normal transactions because there is no fee
- Solutions: Prioritize the processing of mining involved transactions first
- Change location : Block structure