NEXTY: A CONSENSUS TO GET ZERO TRANSFER FEE AND INSTANT TRANSFER BLOCKCHAIN

THANH DAO & HA DANG
CO-FOUNDER/CTO @ NEXTY PLATFORM
THANHDAO@NEXTY.IO / HADANG@NEXTY.IO

Abstract. This is the detail technical paper of Nexty Platform, focusing on describing the operation of a consensus protocol called Proof of Foundation. Proof of Foundation was inspired by the Proof of Authority introduced by Szilágyi [2017] with improvement from Nexty Platform by introducing a new confirmation system named **DCCS** - **D**ual **C**ryptocurrency **C**onfirmation **S**ystem to achieve a decentralized system and bring a highly incentive system for blockchain maintainers.

1. Introduction

In addition to NTY coin-base, DCCS has the secondary token named NTF. NTF is used for authorizing an account to become the maintainer/sealer of the confirmation system described detail in section 2, as well as to calculate reward for block sealer. NTF token has total supply of $10,000,000$ and will be distributed for the first $100,000,000,000$ NTY holders that joined in "Smart Staking" program described detail in the white paper of Nexty [2017]. In the other words, NTF will be rewarded to the pioneers having a clear vision and a strong believe in Nexty Platform in the future. That's the reason why we call the new consensus protocol as *"Proof of Foundation"*. The power of Nexty Platform, however, is not belong to NTF holders because they could be voted down by Nexty community if community found that NTF holders did any bad behavior of cheating, malicious or hand-shaking to make the network become centralize chain as well as using out-of-date source code. As a result, NTF holders have the only role as block sealer for Nexty blockchain and governed by NTY community via decentralize voting system.

2. How to authorize an account to become a block sealer

The system will set up a configuration parameter to define the minimum value, $\mathbf{50,000}$ NTF, that an account need to deposit into Nexty governance smart contract to become a block sealer. Nexty will build and develop a governance smart contract, which allow NTF token holder, having enough token, to grant another account called "executing-account" to become the block sealer by setting "authorized-sealer" state in the smart contract with the value equal to address of NTF holder. If an address already has an "authorized-sealer" value, it can not receive authorization from another NTF holder until the NTF holder has made the withdrawal it from "authorized-sealer". The "authorized-sealer" list will be determined at the check point block of each epoch by reading canonical state from the smart contract at that point of time. If in the last three epochs, any "authorized-sealer" does not perform at least one sealing activity, the "authorized-sealer" value will slash by updating the state of the smart contract at the checkpoint block number of the next epoch and of course that "authorized-sealer" will not be involved in the next epoch until it will be authorized to

become "authorized-sealer" again via community voting dapp. We will introduce more detail about this on Nexty governance section later 4.

3. Block sealing mechanism

The DCCS block sealing mechanism will be implemented as in following steps. Firstly, the sealers will be numbered from 0 to $n-1$ (called "sealing-id") randomly at the beginning of each epoch; in which, n is the number of registered "authorized-sealer" and is determined by the state of the governance smart contract at the canonical chain from *checkpoint block number* of the corresponding epoch. To ensure the randomness of "sealing-id", the numbering is calculated by a hash, $\boldsymbol{\xi_k}$, as following formula when starting a new epoch.

$$\xi_{\mathrm{k}} \equiv \mathtt{KEC}(\mathbf{block}, \boldsymbol{\Lambda}_{\mathrm{k}}) \tag{1}$$

block: is the canonical block number from the *checkpoint block number* of the epoch. a.k.a *checkpoint block number* - 7

Λ_{k}**:** is the address that the NTF holder has set as "authorized-sealer" in the governance smart contract.

`KEC`**:** is the **SHA-3 Keccak-512** hash function of any input.

After that, the "sealing-id" of each authorized-sealer will be taken by the position of *sealing hash*, $\boldsymbol{\xi_k}$, in the array $(\xi_0, \xi_1, ..., \xi_{\mathrm{n}-1})$, ascending ordered by the *sealing hash*.

To ensure performance of the system, the "sealing-id" of the all authorized sealers will be snapshot only once from the smart contract state at the canonical chain from *checkpoint block number* of each epoch and stored in the local database as well as in **lru cache** of each node.

If sealing node is not in recent sealers. The node will determine whether it's the in-turn sealer for the next block or not, according to the following formula:

$$\sigma_{\mathrm{k}} \equiv (\nu - \mathbf{block}) \quad \mathbf{mod} \quad \Pi \tag{2}$$

ν**:** is the current block number that is being sealed.

block: is the canonical block number from *checkpoint block number*, of the current epoch.

Π**:** total number of authorized sealer reading from smart contract state at the canonical block from *checkpoint block* of the current epoch.

If the remainder σ_{k} equal to "sealing-id" of the node, then that node has in-turn sealer and seal block immediately with highest *difficulty*. Otherwise, nodes will wait a period of time, ψ_{k}, which is calculated by the formula:

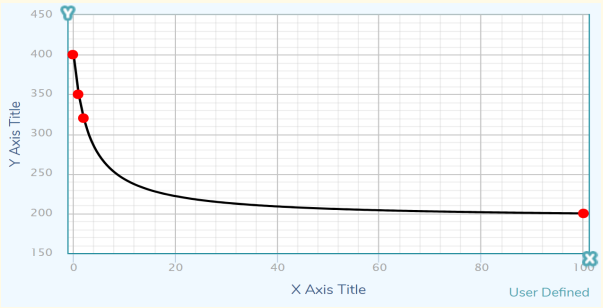
$$\alpha = 001.387978000 \tag{3}$$
$$\beta = 000.002313279 \tag{4}$$
$$\gamma = 000.004626590 \tag{5}$$
$$\delta = 499.999400000 \tag{6}$$

$$\psi_{\mathrm{k}} \equiv \sum_{i=1}^{\zeta_{\mathrm{k}}} \mathbf{floor}\left(\frac{\alpha}{\beta * i + \gamma} + \delta\right) \tag{7}$$

The coefficients of the above formula are assumed as following. The sealing node having "sealing-id" equal to σ_{k}, calculated by formula (2), will have the priority to seal block before 400ms, the next sealing node will have the priority to seal block before 850ms, then 820ms,... till the final sealing node with approximately to 500ms. Its graph corresponds to the function $\mathtt{y} = \frac{a}{(b*x+c)} + \mathtt{d}$

To estimate values for the parameters, we use curve fitting method to find relatives coefficient in the website My Curve Fit with the variance [1] ≈ 1.

[1] R^2 is 1 minus the ratio of sum of the squares of the residuals divided by the sum of the squares of the differences between Y fit and the mean Y value



The difficulty of sealing block for n sealers will be calculated in the formula (8) as below

(8)

$$d(i) \equiv \begin{cases} n & \text{if } \textit{sealer} \text{ is in-turn} \\ n - k_i & \text{if } \textit{sealer} \text{ is out-turn} \end{cases}$$

k_i is the relative position of *out-turn* sealer i^{th} to the position of *in-turn* sealer in the current sealing order list.

4. Nexty Governance

This section will describe detail implementation of registration smart contract deployed at address **0x12345** using for consensus system.

4.1. Terminology.

sealer: a participant in the Nexty consensus system. You can become one by depositing DEPOSIT_SIZE NTFs into the Nexty mechanism.

active sealer set: those sealer who are currently participating, and which the Nexty mechanism looks to seal blocks and other consensus objects.

withdrawal period: number of epochs between a sealer exit and the sealer balance being withdraw-able.

coinbase: the account address that sealer will use to sign when sealing block.

registration contract: a smart contract for handling Nexty consensus mechanism.

4.2. Constants.

Constant	Value	Unit
BLOCK_TIME	2	seconds
EPOCH_LENGTH	3,000	blocks
DEPOSIT_SIZE	50,000	NTFs
WITHDRAWAL_PERIOD	30,000	blocks

4.3. Sealer status codes.

PENDING_ACTIVE = 0: Sealer deposited enough NTFs into registration contract successfully.

ACTIVE = 1: Sealer send request to become a sealer and added into activation sealer set successfully.

PENDING_WITHDRAW = 2: Sealer send request to exit from activation sealer set successfully. Sealer casted out of activation sealer set.

WITHDRAWN = 3: Sealer already withdrawn their deposit NTFs successfully. They can only make withdrawal after withdrawal period.

PENALIZED = 127: Sealer marked as penalized node (update by consensus or voting result via dapp) and cannot become active sealer and cannot withdraw balance neither.

4.4. Nexty chain registration contract.

A registration contract is added to the Nexty chain to deposit NTF and activation sealer set management. It has some below basic functions:

4.4.1. *Sealer Operations.*

Deposit: Transfer the NTF from token holder to registration contract. Sealer might have to approve contract to transfer an amount of NTF before calling this function.

Join: To allow deposited NTF participate joining in as sealer. Participate already must deposit enough NTF via **Deposit** function. it takes **coinbase** as parameter.

Exit: Request to exit out of activation sealer set.

Withdraw: To withdraw sealer's NTF balance when they already exited and after **withdrawal period**.

4.4.2. *Sealer Record.*

Coinbase address: sealer's coinbase address to sign when sealing block
Balance: the total amount of NTF that sealer has deposited
Status: Sealer status code
Requested Block Number: The block number at which sealer send request to exit out of activation sealer set.

4.5. **How to become a sealer.**

(1) A participant should deposit at least DEPOSIT_SIZE NTFs into registration contract, a.k.a the network have maximum 200 candidate nodes can become a sealer. Sealer status will be set to PENDING_ACTIVE.
(2) A participant need to set their **coinbase** to seal the block and receive reward on their equivalent NTF token holder address.
(3) A participant call registration contract method to become sealer. Sealer status be updated to ACTIVE.
(4) At next checkpoint of the epoch, sealer request be cast and the sealer be added into consensus **activation sealer set**.

4.6. **How to exit from activation sealer and NTF withdrawal.**

(1) Sealer call registration contract method to request for exiting from activation sealer set and sealer status be updated to PENDING_WITHDRAW but still in the activation sealer set on the current epoch.
(2) At next checkpoint of the epoch, sealer exit request be cast and the sealer be removed out of **activation sealer set**.
(3) Sealer can call register contract method to withdrawn their deposited NTF after a withdrawal period. The sealer status be changed to WITHDRAWN after sealer withdraw the NTFs successfully.

4.7. **Sealer authorization.** When a node starting with *–mine* option then they need to unlock the unique **coinbase** account setting in the activation sealer set. If the **coinbase** was not found in the current epoch **activation sealer set** then unauthorization error will be throw and it cannot start mining but they still can continue to sync block preparing for sealing later.

4.8. **Penalized.** At the checkpoint of each epoch, consensus will collect the data from the chain to find out any sealers that didn't seal any blocks 3 epochs in a row. Then to ensure the network healthy, those sealers will be removed out of the **activation sealer set** in the new epoch and mark sealer's status as PENALIZED. Because, offline sealers may due to the nature of unstable global network and dis-continue infrastructure service that out of control of the sealers. Then all PENALIZED sealers then can be unlock the slash after voting up by NTY community via a specific voting smart contract as well as dapp.

5. Block sealer reward calculation

Reward will be added immediately into sealer's account balance after sealing a block. The reward for a block is equal to the number of rewards per year divided by $\mathbf{15,768,000}$, the expected number of block within one year (initial configuration of block time is 2 seconds). The amount of rewards of each block sealing in a specific year is calculated in the following proportions:

Year	% of total supply	Reward/Blk
1	10.000	1,141
2	5.000	627
3	2.500	329
4	1.250	169
5	0.625	85
6	0.500	69
...	0.500	...
...	0.500	...
n	0.500	...

6. Anti-SPAM

6.1. **Most Frequency Use.** In Nexty blockchain, we allow end user to perform transaction with zero fee. Therefore, we introduce a mechanism to priority transactions base on some other factors other than using only the **gasprice**. The first factor is similar to stamina, each account will have a **most frequency use** state. whenever, transaction from account to include in the block then the **most frequency use**, Δ_{new}, will be update as following formula.

(9)
$$\Delta_{new} = \Delta_{old} + max(1, \frac{(block - \Delta_{old}) * GasUsed}{3 * 21000})$$

block: is the block number that transaction was include.

GasUsed: gas used by the transaction

Ideally, the higher **most frequency use** value, the less priority of the account's transaction in the transaction pool. And each time an account perform the transaction, **most frequency use** value of that account will be increase; as a result, the next transaction of the same account will be less priority. And to avoid spam by new account, the **most frequency use** of a new account will be the block number of the account's first transaction.

6.2. **Transaction Parity.** Along with **most frequency use** above, we introduce a new factor to decide the priority of the transaction called **parity** and will be calculated as below

(10)
$$\rho_t = \Delta_{act} + \frac{TxGasLimit}{21000} - \frac{TxGasPrice}{ParityLimit}$$

Δ_{act}**: most frequency use** value of the account when perform transaction

TxGasLimit: transaction gas limit, the higher gas limit transaction, the less priority

TxGasPrice: transaction gas price, can be input arbitrarily by user. User can put higher gas price if they want to make transaction be confirmed faster.

ParityLimit: parity limit be configured by miner, the higher parity limit, the less transaction gas price effect into the overall parity value, ρ_t

Overall, a transaction was include in the miner's txpool with priority base on the value of transaction parity calculated in formula (10) then even though Nexty allow user to send tnx with zero fee but sealer's pool can arbitrarily setting parameters to utilize the sealer's resource for anti spamming transaction and ensure normal user's transaction to be proceed without delay.

6.3. **Transaction Proof of Work.** In the near future, when the Nexty chain will be fulfilled or be spammed by unmeaning transactions, we will introduce a mechanism to avoid those kind of problem by enforcing user to send a proof of mini work along with the transaction, the transaction with valid proof will be sealed in the block with highest priority. To make a better user experience and adoption, we will develop this feature in user wallet; when the transaction confirmation is too slow, wallet will allow user to take action to do transaction proof of work on their wallet before sending transaction.

References

Nexty. Nexty platform white paper, Dec. 2017.

Péter Szilágyi. Clique poa protocol and rinkeby poa testnet, Mar. 2017.