Nexty Consensus Protocol 1.1

NOTICE: This document is a work-in-progress for implementers. It's subject to change until being marked as FINAL version and reflects recent spec changes and takes precedence over the nexty yellow paper.

Introduction

This document make some update & detail implementation of nexty yellow paper; especially detail implementation of **registration smart contract** using for consensus system.

Terminology

1. **Sealer** - a participant in the Nexty consensus system. You can become one by depositing 100,000 NTF into the Nexty mechanism.
2. **Active sealer set** - those sealer who are currently participating, and which the Nexty mechanism looks to seal blocks and other consensus objects.
3. **Slot** - A blocks period that equal to activation sealer set size.
4. **Withdrawal period** - number of epochs between a sealer exit and the sealer balance being withdrawable.
5. **Coinbase** - the account address that sealer will use to sign when sealing block.
6. **NTF** - ERC20 token that sealer need to deposit into registration contract to become a sealer.
7. **Registration contract** - A smart contract for handling Nexty consensus mechanism.

Constants

Constant	Value	Unit	Approximation
BLOCK_TIME	2**1 (= 2)	seconds	
EPOCH_LENGTH	3*10**4 (= 30,000)	blocks	~17 hours
DEPOSIT_SIZE	10**5 (= 100,000)	NTFs	~100 sealers

Sealer status codes

Name	Value	Explanation
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WITHDRAWN	3	Sealer already withdrawn their deposit NTFs successfully.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.

Consensus

1. **Nexty Token Foundation - NTF**

- Type: ERC20
- Total supply: 10,000,000
- Decimals: 18
- Name: Nexty Token Foundation

2. **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:
    - **Deposit**: Transfer the NTF from token holder to registration contract. Sealer might have to approve contract to transfer a 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
    - **Withdrawn**: To withdraw sealer's NTF balance when they already exited and after withdrawal period.

- 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
- Sealers mapping
    - A mapping containing all sealers' record: ***mapping***(holder => [Sealer Record])
- Activation sealer set
    - A mapping containing **activation sealer set**: ***mapping***(coinbase => holder)
    - Both **cointbase** and **holder** should be unique.

3. How to become a sealer

- A participant should deposit at least **DEPOSIT_SIZE** NTFs into registration contract, a.k.a the network have maximum **~100** candidate nodes can become a sealer. Sealer status will be set to PENDING_ACTIVE.
- A participant need to set their **coinbase** to seal the block and receive reward on their equivalent NTF token holder address.
- A participant call registration contract method to become sealer. Sealer status be updated to ACTIVE.
- At next checkpoint of the epoch, sealer request be cast and the sealer be added into consensus **activation sealer set**.

4. How to exit from activation sealer and NTF withdrawal

- 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.
- At next checkpoint of the epoch, sealer exit request be cast and the sealer be removed out of **activation sealer set**.
- 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.

5. 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.

6. **Sealer selection & block sealing**

- At epoch checkpoint, node will read registration smart contract to get the current activation sealer set and store the set into local snapshot cache and database. ***N.B:*** This **activation sealer set** is not updated in the whole current epoch. Any update of activation sealer set will only be cast on the next checkpoint and apply to the next epoch.
- Activation sealer set will be order by the ordering hash of combination between previous block hash and sealer's **coinbase**.
- Sealer's in-turn/out-turn will be calculated as mentioned in yellow paper.
- When a sealer is in-turn, it can seal block and submit block to the network otherwise it need to wait a period of time estimating by a formula as mentioned in yellow paper.

7. **Reward**

- Reward will be added into sealer's account balance after sealing a block by **slot** basis of current epoch. A sealer only receive one time reward in each slot even they seal more than one block in the same slot. Reward amount still calculated as mentioned in the yellow paper.

8. **Penalized/Punish & Community/Network voting**
   TBD

9. **Anti-spam & prevent DDOS attack**
   TBD

References

1. Nexty Yellow Paper - https://github.com/nextyio/yellowpaper/blob/master/Paper.pdf
2. Ethereum 2.0 Specifications - https://github.com/ethereum/eth2.0-specs
3. Flexible-PoS-Rewards by VietLQ