TDMPC Protocol

Time Division Multiple Proofs Consensus Protocol -

A Consensus Protocol for the next decentralized computer networks

时分多重证明共识协议-次世代去中心化计算机网络协议 v0.52

Michael Yeung, Jan 2018

Abstract: The TDMPC Protocol introduces new frameworks to support a large scale of different blockchains applications. It works with below advantages:

(1) It supports millions of users
(2) It avoids traffic jam due to unexpected incidents (crypto kitty for ethereum)
(3) It reduces the confirmation latency to seconds
(4) It is true decentralized consensus but faster and more efficient
(5) Proof of work as public chain and Proof of Burn for applications, dual consensus
(6) Unique of the Public Chain, Applications and Cloud Computing provides arbitrage free triangle and this makes sure that no one is able to control the whole network
(7) The consensus itself is very easy to understand and easy means safety.

Background:

Bitcoin uses **Proof of Work** and the network chooses the most difficult chain to represent the votes and consensus. PoW can provide the best decentralization of the network as if the network's value is higher than the pricing, it is very easy to have equilibrium by implementing more hashpower to compete. The drawback of Bitcoin's PoW is slow and lacks of Apps supports (turing incomplete).

Ethereum tries to address Bitcoin's problem by increasing the settlement speed with turing complete script languages. Ethereum can have a better DApps support than Bitcoin. However Ethereum also has two problems cannot address: (1) **Efficiency Problem.** The miners have not much incentives to execute large scale or data intensive projects. The data storage in Ethereum network is very expensive and some reports indicate that the "operator" is 400 millions more expensive than AWS (Amazon Web Services). (2) **Discrimination Problem:** Because Ethereum is an open platform for all DApps like Android, it is difficult for the network to

anticipate the upcoming traffic (Crypto Kitty Incidents). Also it gives the miners to discriminate some DApps from the others.

Proof of Stakes (PoS) is given out to address the PoW's efficiency problem. Whoever has more tokens, has higher chance to mine the blocks. It means that whoever has more tokens, has more responsibility of the network and is willing to pay more for the system. However, since the miners do not need to spend too much resources on mining, when the network has more value than pricing, it can attracts outsider to attack with much higher hashpower but without any tokens. Unlike PoW, the miners within the network does not have much hashpower to defend. And most of the users will choose to fork to max their best interests. This is called "**Attack of no Statke".**

Delegated Proof Statke (DPoS) was developed to counter "Attack of no stake". In EOS.io, the tokens are the votes. The nodes use their tokens to vote 20 delegators + 1 random selected delegator. Within a certain period, the one of the 21 delegators takes turn to be the "president". Like electing president in USA, EOS.io even allows nodes to vote a constitution to manage the possible fork event or fixing bugs. DPoS can solve the "Attack of no Stake" problem. Also since the 21 delegators are likely to be stable, the owner of EOS.io is more willing to invest more on the resources (hardware, bandwidth, storage and CPU). However, although there are 21 delegators, EOS.io cannot stop one monoplayers to split 20 IDs to control the network initially. Once the network is controlled, it is impossible to challenge such "**dictatorship**".

TDMPC uses two asynchronous consensuses to address above problems. One uses Proof of Work (PoW) to generate the "wood". The woods are the fuels to be used and burn out （Proof of Burn) to generate tokens for different blockchain applications. Therefore,

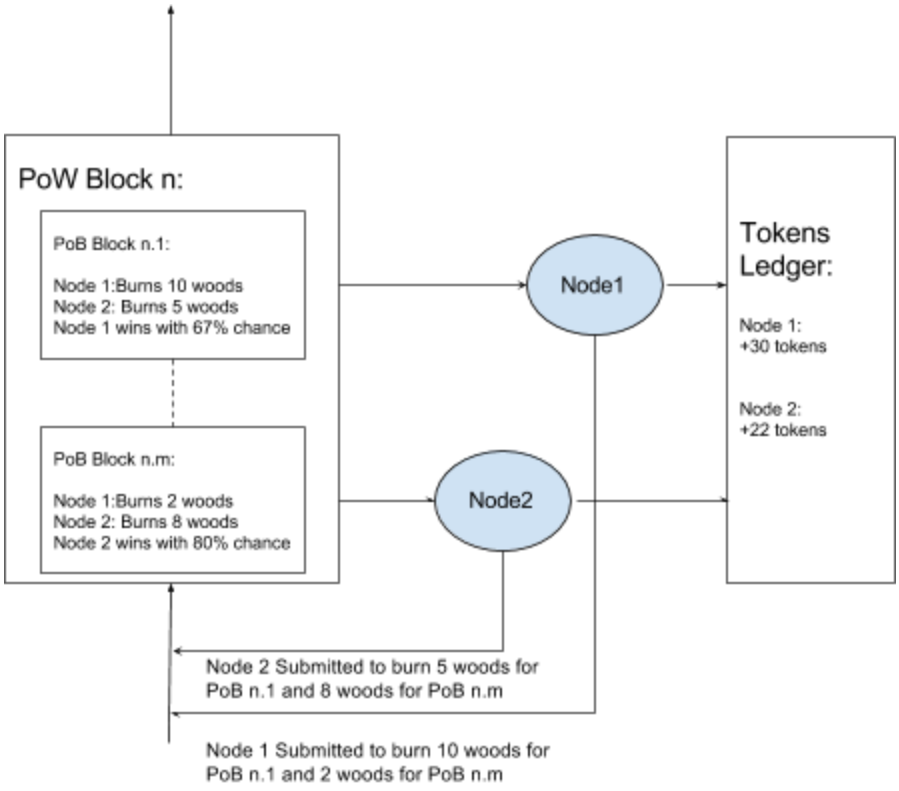
(1) If the network has more value than pricing, the outside hashpower can compete with the network to have equalize the pricing. This proves "**decentralization**" and avoid the "**dictatorships"**.
(2) Because the woods are burn out to generate the tokens, the network needs to keep a certain amount of the hashpower. Also since the PoW and PoB is asynchronous, this give the network some time to respond the attack. Moreover,the network can play some strategy by spending "woods" into different periods to resist the attack. This prevents the **"Attack of no Stake**".
(3) Unlike PoW or PoS or DPoS, the tokens in TDMPC is closed loop. The tokens for the miners come from the users who pay the tokens for the DApps. It means the miners cannot have the fixed awards. Therefore it gives the miners to invest enough resources for the high quality DApps to avoid the **discrimination problems**.
(4) At last, since the woods is the ticket to gain the rewards, and the quality of the hardware represents the amount of the rewards itself. Any new hashpower joining the network has to equip enough hardwares as well.

Timestamp Server:

(1) PoW time slot is set to be a nonce “n” for every a certain amount of the time
(2) PoB time slot is set to be a nonce “m” for each PoW time slot

General Process:

(1) Each node announces how much woods to be burn for incoming each PoB Block n.m to the PoW network
(2) Nodes who wants to earn woods, solves the hash problem to earn the PoW block n
(3) PoW block n obtains the submissions from each node for the PoB process
(4) PoW block n calculated the odds of each PoB n.m block from the woods submission and announces the winner for each PoB n.m block.
(5) PoW block adds a certain amount woods to the PoW winner
(6) Until the next PoW block, the network is carried by the nodes following the orders list which is announced by the current PoW block
(7) Each PoB winner has the right to run applications and earn tokens and manage the Tokens Ledger



PoW and generating the woods

(1) Each node solves hash problem to earn the rights to execute the PoB requests and a certain amount of the woods
(2) The Network agrees on the chain with the most woods burn (the most consensus)
(3) The wood is used for fuel generation, therefore PoW it is not necessary to be very fast.

PoB and generating the tokens

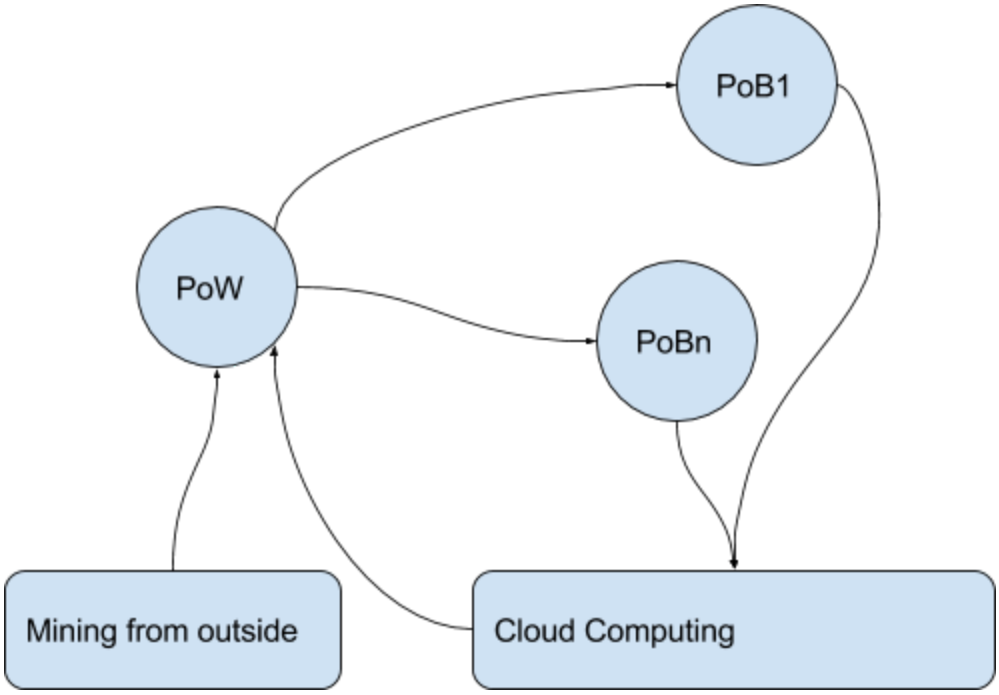
(1) The wood is used to generate any token by Proof of Burn
(2) For each PoB block, node burns out a specific amount of the wood for tokens with a certain probability (the more woods burn, higher the chance to get the tokens)
(3) Whoever wins mines the block of the application and gets the tokens as the rewards
(4) The “burns” is a specific term in smart contract, usually refers to “destroy”
(5) The amount of the wood burn should be spent carefully to maximize the node’s benefits. Because different application / token shall represents different value

Cloud Computing Mining (optional) is used for working out the PoW problem by consuming the tokens. The price of the tokens shall be decided by the market

Hashpower Outside the Network is the force from outside to have the equilibrium / competitions with the network. If the network has more value than the Cloud Computing Mining power, i.e. the wood is controlled by very few nodes, the outside power shall be able to compete with them.

Empty Block Attack for PoW process and hard forks: There is a possibility for a greedy PoW winner, only includes the node itself but excludes all other PoB submission to take advantage of the making the PoB list. This is so called empty block attack. The attack is not working because if another PoW (2nd winner) includes everyone , the other notes should choose the chain with most woods burn, i.e. the 2nd winner i/o the 1st winner. The first winner’s block will be abandoned, so called hard fork.

Empty Block Attack for Token ledgers: There is no incentives for PoB winner to do empty block attack as this node has to burn wood first. And in our consensus, you can only earn tokens by running applications and earn fees. You can’t make fees by creating empty block in PoB.



Proof of Correctness: Below proves the correctness of our consensus is more efficient and stable than normal PoW.

We define below:

$$M = m_1 + m_2 \sim (1)$$

$m_1 \sim Node's\ total\ resources\ invested\ in\ mining$
$m_2 \sim Node's\ total\ resources\ invested\ in\ system\ to\ run\ applications$
$M \sim Node's\ total\ resources\ invested\ in\ mining\ and\ system\ to\ run\ applications$

Therefore the each node's profits from mining and running applications is below:

$$Return = \frac{m_1}{P} \cdot (a + \frac{m_2}{C} \cdot b) \sim (2)$$

$P \sim total\ mining\ resources\ within\ the\ system$

$C \sim total\ applications$
$\frac{m_1}{P} \sim expected\ mining\ hit\ ratio$
$a \sim fixed\ rewards\ from\ mining$
$b \sim\ rewards\ from\ running\ applications$

It is also easy to have below equations:

$m_2 \leq C \sim$ （3）
$m_1 \ll P \sim$ （4）

According to (1) and (2), we have

$$Return\ =\ -\frac{b}{P \cdot C} \cdot m_1^2 + (\frac{M \cdot b}{P \cdot C} + \frac{a}{P}) \cdot m_1$$

then if set m_1 is an independent variable and others are constants, the formula of return is a quadratic equation with negative second order coefficient.

There is a solution of m_1 for the node's max return giving that:

$$m_1 = \frac{M}{2} + \frac{C}{2} \cdot (\frac{a}{b})$$

It is easy to find out that the m_1 can max the return is independent from the total network calculation P.

Let's take a/b as another independent variable, which represents the ratio of the fixed rewards and rewards from running applications. Recalls the formula (4), and a must be bigger or equal than 0, we can have below condition:

$$0 \leq \frac{a}{b} \leq \frac{M}{C}$$

If sets a = 0, which means it gives no fixed incentives. The incentive is 100% coming from running the applications. Therefore, when

$\frac{a}{b} = 0$,

$m_1 = \frac{M}{2}$

It means that if giving no fixed incentives for mining, node will allocate half of the resource to max the node's benefits, for sure and stable.

But, when taking the max “a/b” from 0 to “M/C”, $m_1 \; increases \; to \; M$, this means when adjusting the rewards ratio to M/C, the node is motivated to use 100% resources on mining rather than spending a cent on running applications. This is usually the case for PoW network. It also explains why within PoW network it is difficult to run a scale application.

It is not difficult to see that TDMPC is a consensus to adjust a/b = 0 which provides a much better and more stable network to run applications than normal PoW. Proof is completed.