Hybridicum: A Hybrid Blockchain Consensus Protocol

Alexander Chepurnoy

Abstract

Keywords: Blockchain, Decentralized consensus, Proof-of-stake, Peer-to-peer networks, Proof-of-Work

1. Introduction

TODO: terminology and notations

2. Interactive Proof-of-Stake

3. Properties For A New Blockchain Consensus Protocol

We formulate following properties of a new blockchain consensus protocol we wish to achieve.

1. Block generators in Proof-of-Work are incentivized to contribute to one chain only. In opposite, block generators in Proof-of-Stake are incentivized to contribute to as many chains as possible. We would like to get chain selection from a tree enforced in an elegant way.
2. Proof-of-Stake provides an incentive to run a fullnode.
3. We are willing to spread transaction fees amongst few parties. Otherwise a block generator could include her transactions for free. There are some attacks linked with that, for example [**?**]

Email address: kushti@protonmail.ch (Alexander Chepurnoy)

4.

5. The Hybrid Consensus Protocol

With Interactive Proof-of-Stake miners still have incentive to contribute to a multiple chains. Also grinding attacks are probably possible(???).

To overcome the shortcomings aforementioned, the idea of ours is to use both Proof-of-Work and Interactive Proof-of-Stake to secure a blockchain system.

We use Proof-of-Stake blocks(*containers*) to carry transactions. We use a Proof-of-Work block(*deciders*) as a random beacon[**?**] and also as a voting mechanism to enforce some chain choosing from a tree.

A Proof-of-Work block contains some *puz* value[**?**]. Consider $hash(puz \cup pk)$ has length of 30 bytes at least, so we take first 30 bytes of it and consider those bytes as 10*3 m values to generate tickets for next 10 blocks container blocks. For example, for first 3 bytes up to three tickets for a first block could be generated.

So if online miners set is known next 10 container blocks generators are also known, but network interaction is needed to generate those blocks.

Referencing rules

1. Each container block refers to a previous container block and a parent decider block as well.
2. Each decider block contains references to a parent decider block and last seen container block. Last seen container block must be a descendant of a parent's last seen container block.

Rewarding rules

1. For a container block, transaction fees are divided equally amongst ticket generators.
2. For a decider block, reward is $C * \frac{\delta s_{cur}}{\delta s_{prev}}$, where δs_{prev} is blockscore increasement fixed by previous decider and δs_{cur} is current increasement.

I have no any conflict of interest to declare.