This is a specific proposal on how to implement some of the “PoW in PoS protocols” described in Delayed Protocols. Shout out to @drstone for the inspiration for this scheme.

NOTE: there are probably better ways of doing this, but this is what fell out of trying to shove PoW into the CBC Casper framework. The better way might be something like “provide a PoW on some random number and get new coins,” but I think this is more fun

One main criticism of PoS consensus algorithms it that of the “initial distribution question.” For example, imagine Bitcoin had started with PoS instead of PoW - Satoshi would have just ended up with all the coins, and it all would have ended there. So here’s an attempt to solve the initial distribution question w/in the current PoS frameworks we have - namely Casper FFG, CBC, and Tendermint.

In all the above systems, we have some set of validators V

, and a function W : V \to \mathcal{R}+

, where W

is the weight function. Usually, we think about defining a validators weight as the number of tokens they have staked. However, it may be possible to think about defining these weight maps in other ways. For example, validators (really, miners) could do non-outsourceable PoW on their own address, where their weight is the PoW on their address.

If these miners do more PoW on their address, then they have to “deposit” this PoW in the same way we handle validator set changes currently within the above-mentioned protocols (usually: it has to happen under consensus).

From there, the miners run any of the consensus algorithms mentioned above as usual: in Tendermint, making blocks and pre-voting + pre-committing; in FFG: making blocks and voting; in CBC: making blocks. This means we don’t have a “traditional PoW” protocol, in the sense that the forkchoice is not a longest-chain rule. Also, we are required to make a weak subjectivity assumption as well.

With this, we can have a protocol that avoids the initial distribution question, seemingly as much as regular PoW does.

However, there’s one major issue with the above scheme. Essentially, as soon as a miner decides they don’t want to mine/validate anymore, all the PoW they’ve done on their address is worthless. This is because the PoW is only worth the discounted future rewards it allows the miner to receive (which is 0 when they no longer plan on mining/validating). This is different from PoS, where their coins have value even after they’ve stopped validating (and so it hurts them to be slashed).

To fix this (and where the delayed protocols come in), we just have to delay giving the validator their rewards until the end of the withdraw period. Now, they can’t equivocate w/out losing their rewards from their time as a miner.

There are two problems remaining, one is due to the nature of delayed rewards, and the other due to the nature of PoW:

- Delayed rewards might make it harder for validators/miners to exist, as these validators/miners may have upkeep costs they need to pay in real time (and they can’t exactly borrow against their future rewards

)

- This scheme is less secure than a PoS protocol that pays the same amount of rewards - as there’s significantly less at stake for some validator.