Stake-Based Merged Consensus

Special thanks to @adlerjohn for the discussions that led to this post and for the revision!

Preliminary Readings

- Building Scalable Decentralized Payment Systems
- Minimal Viable Merged Consensus

Overview

We present a variation of merged consensus that is secured through PoS instead of PoW. In our scheme, we do not need a separate token and a separate set of validators/block producers as presented in earlier PoS-secured sidechains. This spec works irrespective of the sybil-resistance mechanism used on the main chain.

Assumptions

Network Model

We assume a partial synchronous network model. This follows from the fact that the main chain also follows a partial synchronous network model.

Adversarial Model

We assume a well-resourced adversary. In other words, an adversary that can afford to burn lots of money.

Honest Majority Assumption

We make the assumption that the block producers on the sidechain form an honest majority. We only need to assume an honest majority for liveness. We discuss in detail in a later section why we get safety without this assumption.

Randomness Beacon Assumption

We assume there exist a randomness beacon that can uniformly subsample committees of block producers. In practice, this beacon can be provided by the main chain in some form.

Stake-Based Merged Consensus Spec

Most of the minimal viable merged consensus spec applies in this case except for the validator registration, fork choice rule, data availability and, finalization. We go in-depth into each of these in the follow sections.

Registration

In order to solve the nothing at stake problem, validators validating on the sidechain need to be penalized whenever they misbehave. So, the validators need something at stake in order to be incentivised to behave properly. This is a bond, which is discussed in the next section.

Validators on the sidechain are a subset of the validators on the main chain. In order to enforce this restriction, main chain block producers must provide a hash of side chain blocks in a main chain block

Fork Choice Rule

Here, the validators use a GHOST-based fork choice rule in order to avoid the pitfalls of Nakamoto consensus (longest chain) when applied to PoS. The fork choice rule is defined by a smart contract on the main chain as in PoW-based merged consensus. However, the main difference here is that forks are possible due to the properties of GHOST-based fork choice rules and for each block that is proposed, a bond B

is provided by the side chain block producer.

Data Availability

Since we now have a consensus mechanism with forks, we can change the data availability guarantees of our merged consensus spec. Now, we can assume optimistic execution on the sidechain and have slightly weaker data availability assumptions. Instead of posting all data on chain (or in shards), we can instead use proofs of custody to prove that certain pieces of data are held. This would enable us to use fraud proofs. A practical scheme would be to have a pre-compile contract on the main chain that runs data availability checks. Then, we can loosen our data availability and honest majority assumptions on the sidechain. We no longer have potential safety violations.

Finalization

In the original merged consensus spec, the strongest form of finalization that could be provided was probabilistic finalization. This is because the main chain was assumed to use PoW, which can only provide probabilistic finalization. In our stake-based scheme, we are able to provide deterministic finality on the sidechain. Deterministic finality gives us the strongest possible form of finality that blockchain-based systems can provide. Any byzantine finality gadget suffices. The rewards and penalties for finalization are dependent on the finality gadget in use. We leave this to future work.

Future Work

Future work consists of relaxing certain assumptions, determining appropriate GHOST-based fork choice rules and a more detailed economics analysis of the incentives in this scheme.