Here is a scheme that allows light clients for Casper with very fast syncing properties without having to even bother implementing the Casper FFG consensus logic; it uses cryptoeconomics.

Validators have the ability to sign messages, of the form [epoch, checkpoint_hash, validator_index, sig]

. An additional slashing condition is added where a message of this form that does not have the correct checkpoint hash for the epoch it specifies means that the validator that signed it can be slashed.

A client that was online at some previous point, less than the withdrawal period in the past, and which has C as its latest known authenticated finalized checkpoint, can use Merkle proofs to download the validator set active at C, then send the network a message asking for checkpoint messages from those validators for the most recent checkpoint they are willing to attest to. If some checkpoint gets more than some threshold (eg. 10% of validators) attesting to it, then the validator accepts that as its new authenticated checkpoint. The client rebroadcasts these messages to make sure that if they are incorrect, the validator that creates these messages gets slashed.

This allows light client resyncing that’s not only not slower, but actually much faster

, than proof of work.