The papers Casper the Friendly Finality Gadget and Combining GHOST and Casper have proved the safety: there is only one canon chain in any view.

But they assume there is only one global static validators set, while in the implementation, the validators are dynamic, which results in: at a specific time, different validators can see different validator sets instead of a global validator set.

In this case, we may have multiple canon chains without 1/3 validators get slashed. A simplified example is shown below:

[

casper_multiple_finalized_blocks

617×727 16.8 KB

](https://ethresear.ch/uploads/default/original/2X/2/29e11137df9a41b0490b88f13c7e0e4552551892.png)

The blocks are the epoch boundary blocks. All the arrows are justifications(since 2/3 validators vote). v2, v3 joined(or called "activated") at Block<2a>. We can see Block<2a> and Block<3b> are finalized, and they are in different branches, while there is no validator slashed.

It rarely happens in the real situation, but it seems feasible in theory, so maybe it will happen in some tricky cases?

One of a naive solution to fix this I can figure out for now is not to let active validators update util a block is finalized.(Some disadvantages are for example, some validators can not exit in time even when they are not able to be functional etc)