TL;DR

- We propose Casper the Subjective Finality Gadget (SFG)

, a modified version of Casper FFG.

- Casper SFG improves Casper FFG to support a subjective commit rule

, by which clients can finalize a block based on their local assumption on the number of faults.

- We use a mixed fault model for SFG, where fault assumption is analyzed for safety and liveness, respectively.

Background

Some consensus protocols support subjective commit rules

, where clients can finalize blocks based on their parameters. Nakamoto consensus, derived from Bitcoin, allows clients to commit a block subjectively with the "n

confirmation rule" to take a trade-off between security and latency to finality. CBC Casper adopts subjective finality as its design philosophy, and its finality condition is directly parameterized by the fault tolerance. Flexible BFT is a recent proposal in academia, which modifies the traditional PBFT to support subjective commit rules.

As in CBC Casper and Flexible BFT, subjective commit rules are often analyzed in a mixed fault model

, where safety and liveness are proved in different assumptions. This is different from the normal Byzantine fault model, where faulty validators are assumed to attack both safety and liveness. The mixed fault model is the key to make a subjective commit rule because, in the normal BFT analysis, a certain threshold (e.g., 2/3

votes threshold and 1/3

BFT, for asynchronously safe commit) is optimal. Also, the mixed fault model seems reasonable in blockchain use-cases, because incentives around safety failure (e.g., double spend, usually at the cost of punishment) and liveness failure (e.g., censorship) are substantially different.

Subjective finality in Casper SFG

To clarify, we call this proposal Casper the Subjective Finality Gadget (SFG)

.

We can achieve SFG by modifying the finality condition of FFG as follows:

- The votes threshold (by the ratio to the total stake) for a checkpoint to be justified is replaced with a parameter q_r

. * q_r = 2/3

in FFG. Strictly speaking, slightly larger than 2/3

, e.g., 67 out of 100.

- q_r = 2/3

in FFG. Strictly speaking, slightly larger than 2/3

, e.g., 67 out of 100.

- We say a checkpoint is prepared

if more than q_c ~(\ge q_r)

votes for the checkpoint (using the same source).

- A checkpoint gets finalized if it is prepared using its direct parent checkpoint as the source, which is also prepared.
- This rule can be generalized into the [k

-finality rule](https://docs.google.com/presentation/d/1MZ-E6TVwomt4rqz-P2Bd_X3DFUW9fWDQkxUP_QJhkyw/edit#slide=id.g621d74a5e7_0_166) adopted in eth2.

- This rule can be generalized into the [k

-finality rule](https://docs.google.com/presentation/d/1MZ-E6TVwomt4rqz-P2Bd_X3DFUW9fWDQkxUP_QJhkyw/edit#slide=id.g621d74a5e7_0_166) adopted in eth2.

Here, q_r

is an in-protocol parameter, but q_c

is chosen by clients.

Fault tolerance

In our mixed fault model, f_safe

(resp. f_live

) is the ratio of faulty validators who attack safety (resp. liveness). The intersection of these two sets of faulty validators is the Byzantine validator set, whose ratio is denoted as f_Byz

.

The proof of accountable safety is fundamentally the same with the safety proof of FFG, but the fault tolerance is now $f_\mathrm{safe} < q_c + q_r - 1$

. This ensures the quorum intersection

, i.e., if a checkpoint C

is finalized, the number of faulty validators is not sufficient to unlock everyone by making a conflicting checkpoint later than C

justified.

Liveness requires quorum availability

, i.e., $f_\mathrm{live} < 1 - q_c$

.

These fault assumptions are equivalent to Flexible BFT, fundamentally because FFG and PBFT share the two-round finality rule. Therefore, we can refer to this graph modified from Figure 6 in the paper about the possible parameterization.

[

image

864×639 73.3 KB

](https://ethresear.ch/uploads/default/original/2X/c/c254be78ca20f708af13fd908e462edd289f6549.png)

The legend represents the different q_r

values. Flexible BFT focuses on only Byzantine participants or faulty participants who only attack safety but not liveness (called alive-but-corrupt

participants f_abc

), i.e., $f_\mathrm{safe} \ge f_\mathrm{live}$

, $f_\mathrm{Byz} = f_\mathrm{live}$

and $f_\mathrm{total} = f_\mathrm{safe} = f_\mathrm{Byz} + f_\mathrm{abc}$

. Therefore, the shaded gray area representing invalid parameterization exists. We can consider the X-axis as f_live

and the Y-axis as f_safe

.

Closing remarks

This proposal is one of the results of my unpublished paper.

Eth2 can adopt SFG with little modification on the spec, by considering q_r = 2/3

.

Flexible BFT also supports a synchronous commit, as well as the asynchronously safe commit rule we discussed in this post. There are various ways to support synchronous commit in Casper SFG. We will discuss this in a future post.