Suppose that we have an on-chain Casper FFG cycle, which we absolutely want to confirm, and which we want the chain to be aware of, but this cycle would take a much longer time than we find acceptable because of overhead/finality time/decentralization tradeoffs. For example, suppose that we have 32 ETH validator slots, with 10,000,000 ETH validating in total; this would entail 312500 validator entries, which at an overhead of 4 tx/sec would take nearly a day to go through.

We can achieve much faster finality time, in the optimistic case, in practice, as follows. In addition to on-chain Casper FFG messages, validators are free to make off-chain Casper FFG messages. There are two layers of checkpoints: slow checkpoints

(blocks that appear at 1-day intervals) and fast checkpoints

(blocks that appear at 1-minute intervals). An on-chain Casper FFG message contains a (target epoch, checkpoint hash, source epoch) tuple that can only refer to slow checkpoints and slow epochs (ie. in this example, multiples of 1440). An off-chain Casper FFG message contains such a tuple (target_epoch, checkpoint_hash, slow_checkpoint_hash, source_epoch, slow_source_epoch). The slow_checkpoint_hash is the hash at the start of the slow epoch in which the off-chain Casper FFG message is made. source_epoch is the most recent off-chain-justified epoch, and slow_source_epoch is the most recent on-chain-justified-epoch. A justification chain for off-chain blocks can contain on-chain justifications.

We define new slashing conditions:

- No-contradiction: all votes made within one epoch, including on-chain and off-chain, must use the same slow_checkpoint_hash and slow_source_epoch.
- Restricted no-surround, part 1: if an off-chain vote has source epoch o1, target o2 and an on-chain vote source c1, target c2, then it cannot be the case that o1 < c1 < c2 < o2.
- Restricted no-surround, part 2: if one signs an off-chain vote with a source C, then any on-chain votes with a source with earlier epoch than C must reference a target which is a descendant of C.

Safety proof

: suppose incompatible on-chain C1 and off-chain C2 get finalized.

- C2 and C1 are in the same slow epoch. Then, 1/3 of validators get slashed for no-contradiction.
- C2 is in a higher slow epoch than C1. Then, C2 has a justification chain, which must (due to non-contradiction) skip over the slow epoch of C1, which violates restricted no-surround part 1.
- C2 is in a lower slow epoch than C1. Then, C1 has a justification chain, which must eventually have some link with target epoch > C2 and source epoch < C2, where the target is not a descendant of C2. This violates restricted no-surround, part 2.

Plausible liveness proof

: honest validators can always make an on-chain vote with source epoch equal to the most recent known justified slow checkpoint and a target which is a descendant of the most recent known justified fast checkpoint (these must be compatible, because otherwise the fast checkpoint's justification chain would either intersect the slow chain's slow epoch, violating no-contradiction, or skip over it, violating restricted no-surround part 1). Once the target is justified, they can then off-chain or on-chain finalize it.

Fork choice rule

: build on the chain with the highest known justified epoch (could be justified off-chain).

Incentivization

A simple way to incentivize publishing off-chain votes would be to allow all on-chain votes to include a CAS of off-chain votes randomly selected validators. Those votes would later be randomly sampled, allowing the CAS provider to submit a merkle proof at which point the CAS provider and vote signer would both be rewarded.

Light-client verification

Clients could try to download all off-chain messages directly, but this is expensive. They could also use one or more heuristics:

1. Check that there are CASes that sign for enough off-chain votes. Assuming most validators are honest, these are expensive to spoof.
2. Randomly select 500 indices, and download the votes for those indices. Accept a checkpoint as justified if a supermajority of those indices return valid votes that justify the checkpoint.