A Confirmation Rule for the Ethereum Consensus Protocol

Aditya Asgaonkar
Ethereum Foundation

Francesco D'Amato
Ethereum Foundation

Roberto Saltini
ConsenSys

Luca Zanolini
Ethereum Foundation

Chenyi Zhang
ConsenSys

1 Introduction

The goal of this document is to specify a confirmation rule for the current Ethereum consensus protocol, Gasper. A confirmation rule is an algorithm run by nodes that allows them to identify a *confirmed* prefix of the canonical chain, for which safety properties hold; in other terms, it outputs whether a certain block in the canonical chain is confirmed. When this is the case, the block is guaranteed to never be *reorged*, assuming both network synchrony and a certain percentage of stake being honest, i.e., that it follows the protocol.

We first specify a confirmation rule for LMD-GHOST, the component of Gasper that outputs the canonical chain, with proposer-boost, and without considering the finality gadget (or, FFG component) of the protocol. Then, we enhance the rule to account for the influence that the FFG component has on the protocol. In order to do so, we propose a simplification of the *filtering rule* utilized by the fork-choice function implemented by LMD-GHOST, which rules out a certain class of withholding attacks, allowing us to still prove safety of confirmed blocks.

2 Confirmation Rule for LMD-GHOST

Let b be a block, $n-1$ be the slot of b's parent (possibly $slot(b) > n$, if there were missed slots), and let N be the current slot ($N \geq slot(b) \geq n \in \mathbb{N}$). Let β be the fraction of adversarial validators. Moreover,

- Let $\mathcal{W}_b^N$ be the set of validators in the committees from slot n until slot N;
- let $\mathcal{S}_b^N$ be the set of validators in the committees from slot n until slot N that support block b ($\mathcal{S}_b^N \subseteq \mathcal{W}_b^N$);

- let $\mathcal{J}_b^N$ be the set of honest validators in the committees from slot n until slot N ($\mathcal{J}_b^N \subseteq \mathcal{W}_b^N$);
- let $\mathcal{H}_b^N$ be the set of honest validators in the committees from slot n until slot N that support block b ($\mathcal{H}_b^N \subseteq \mathcal{J}_b^N$);
- let $\mathcal{A}_b^N$ be the set of adversarial validators in the committees from slot n until slot N ($\mathcal{A}_b^N \subseteq \mathcal{W}_b^N$); and
- let W_p be the proposer boost value, *i.e.*, the weight which is temporarily granted to a timely proposal. In the current implementation of the Ethereum's consensus protocol, W_p is 0.4 of a committee's weight.

For a set V of validators, let $|V|$ be the *weight* of the set, *i.e.*, the sum of the effective balances, i.e., the balance of each validator that influences its voting power. In particular, we let $A_b^N = |\mathcal{A}_b^N|, S_b^N = |\mathcal{S}_b^N|, W_b^N = |\mathcal{W}_b^N|$, and $J_b^N = |\mathcal{J}_b^N|$. Observe that $\mathcal{W}_b^N = \mathcal{J}_b^N \sqcup \mathcal{A}_b^N$, where $\sqcup$ represent the disjoint union between sets, so $W_b^N = A_b^N + J_b^N$.

Assumption on distribution of adversarial validators. We assume that for every $N \geq n$, $A_b^N \leq \beta W_b^N$. Equivalently, that $\forall N \geq t, J_b^N \geq (1-\beta)W_b^N$. Intuitively, this means that in the union of committees in any consecutive slots, the number of distinct adversarial validators is bounded at a fraction β of the number of total distinct validators.

Assumption on network synchrony. We assume that starting from the time validators cast a vote in the current slot, the network is synchronous with latency Δ lower than the duration between the time when validators cast a vote in a slot, and the end of that slot. Intuitively, this means that starting from the current slot, the votes that honest validators cast in a slot will be received by all other honest validators by the end of that slot. We make no assumptions on the latency of messages sent in prior slots.

2.1 Safety Indicator

Let $p_b^N := \frac{H_b^N}{J_b^N}$. We show in Lemma 1 and Lemma 2, that $\forall b' \in \mathit{chain}(b)\ p_{b'}^N > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ implies safety of b. Still, we cannot directly use this for confirmation, because $p_{b'}^N$ is not observable, since we do not know who the honest validators are. We then introduce the *safety indicator* $q_b^N := \frac{S_b^N}{W_b^N}$, which is very similar to p_b^N, but considers latest messages from *all* validators rather than just from honest ones, and is therefore observable. We show that we are able to detect safety using $\{q_{b'}^N : b' \in \mathit{chain}(b)\}$, because we can check a condition on $\{q_{b'}^N : b' \in \mathit{chain}(b)\}$ which implies that $\forall b' \in \mathit{chain}(b)\ p_{b'}^N > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$, and thus safety.

Lemma 1. *If* $\forall b' \in chain(b)\ p^N_{b'} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W^N_b})$*, then all the honest validators in slot* $N+1$ *will vote for a descendant of block* b*.*

Proof. We proceed by induction proving that for any $b' \in chain(b)$, all the honest validators in slot $N+1$ will vote for a descendant of block b'.

Base Case: b' **is the genesis block.** Obvious as honest validators always vote for a descendant of the gensis block.

Induction Step. We assume that all honest validators in slot $N+1$ will vote for a descendant of the parent of block b'. First, we observe that $H^N_{b'} = p^N_{b'} J^N_{b'} > \frac{J^N_{b'}}{2(1-\beta)}(1+\frac{W_p}{W^N_{b'}}) \geq \frac{(1-\beta)W^N_{b'}}{2(1-\beta)}(1+\frac{W_p}{W^N_{b'}}) \implies H^N_{b'} > \frac{W^N_{b'}+W_p}{2}$. Second, we observe that because of the network synchrony assumption, we know that all the honest nodes in committee $N+1$, by the time they will cast their vote, they will have received all the votes casts by nodes in $\mathcal{H}^N_{b'}$. Observe also that $(W^N_{b'} + W_p)$ represents the maximum total weight that can support any of the children of the parent of b'. The above implies that by the time an honest validator v in committee $N+1$ casts its vote, the weight supporting b' in the view of v will be higher than the weight possibly supporting any of its siblings. This, together with the inductive hypothesis, implies that v will vote for a descendant of b' in slot $N+1$. □

Lemma 2. *If* $\forall b' \in chain(b)\ p^N_{b'} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W^N_b})$*, then* $\forall b' \in chain(b)\ p^{N+1}_{b'} \geq p^N_{b'}$*.*

Proof. Let j^{N+1} be the honest weight in slot $N+1$. Pick any $b' \in chain(b)$. By Lemma 1, we have that $\mathcal{J}^{N+1}_{b'} = \mathcal{J}^N_{b'} \cup j^{N+1}$ and $\mathcal{H}^{N+1}_{b'} = \mathcal{H}^N_{b'} \cup j^{N+1}$ given that any vote supporting b also supports b'. Given that by definition $\mathcal{H}^N_{b'} \subseteq \mathcal{J}^N_{b'}$, we have that $\mathcal{H}^N_{b'} \cap j^{N+1} \subseteq \mathcal{J}^N_{b'} \cap j^{N+1}$. Hence, $\frac{H^{N+1}_{b'}}{J^{N+1}_b} = \frac{H^N_{b'}+x}{J^N_{b'}+y}$ for some $0 \leq y \leq x$ which implies $\frac{H^{N+1}_{b'}}{J^{N+1}_{b'}} \geq \frac{H^N_{b'}}{J^N_{b'}}$. □

Lemma 3. *If* $q^N_b > \frac{1}{2}(1+\frac{W_p}{W^N_b}) + \beta$*, then* $p^N_b > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W^N_b})$*.*

Proof. $\frac{H^N_b}{J^N_b} \geq \frac{S^N_b - A^N_b}{J^N_b} = \frac{S^N_b - A^N_b}{W^N_b - A^N_b} \geq \frac{S^N_b - \beta W^N_n}{W^N_b - \beta W^N_n} = \frac{S-\beta W^N_b}{W^N_b(1-\beta)} = (\frac{S-\beta W^N_b}{W^N_b})(\frac{1}{1-\beta})$. The second inequality comes from minimizing the fraction $\frac{x-z}{y-z}$, where $z \leq c$, which gives $\frac{x-c}{y-c}$. Finally, $p^N_b = \frac{H^N_b}{J^N_b} \geq \frac{S^N_b - \beta W^N_b}{W^N_b}\frac{1}{1-\beta} = \frac{q^N_b - \beta}{1-\beta} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W^N_b})$. □

Lemma 4. *The condition on* q^N_b *in Lemma 3 is strict.*

Proof. Assume $q^N_b \leq \frac{1}{2}(1+\frac{W_p}{W^N_b}) + \beta$. What we want to show is the existence of two sets $\mathcal{H}^N_b$ and $\mathcal{A}^N_b$ such that $p^N_b \leq \frac{1}{2(1-\beta)}(1+\frac{W_p}{W^N_b})$.

To this purpose, pick any $\mathcal{A}_b^N$ such that $A_b^N = \beta W_b^N$ and any $\mathcal{H}_b^N$ such that $H_b^N = \max(S_b^N - A_b^N, 0)$. This corresponds to the possible case where the adversarial weight is βW_b^N and all of this weight supports b up to maximum S_b^N. As a consequence of this, we have that:

- If $S_b^N - \beta W_b^N \leq 0$, then
  $p_b^N = \frac{H_b^N}{J_b^N} = 0 \leq \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$
  as $0 \leq \beta < 1$
- If $S_b^N - \beta W_b^N > 0$, then
  $\frac{H_b^N}{J_b^N} = \frac{S_b^N - \beta W_b^N}{W_b^N - \beta W_n^N} = \frac{S - \beta W_b^N}{W_b^N(1-\beta)} = (\frac{S - \beta W_b^N}{W_b^N})(\frac{1}{1-\beta})$
  Given that $S_b^N \leq W_b^N \left(\beta + \frac{1}{2}(1 + \frac{W_p}{W_b^N})\right)$, then we have that
  $p_b^N = \frac{H_b^N}{J_b^N} = (\frac{S - \beta W_b^N}{W_b^N})(\frac{1}{1-\beta}) \leq \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$

□

Theorem 1 (Safety of confirmation)**.** *If $\forall b' \in chain(b)\ q_{b'}^N > \frac{1}{2} + \beta$, then b is always canonical in the view of honest validators when voting at a slot $> N$.*

Proof. By Lemma 3, $\forall b' \in chain(b)\ q_{b'}^N > \frac{1}{2}(1 + \frac{W_p}{W_b^N}) + \beta$ implies $\forall b' \in chain(b)\ p_{b'}^N > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$. By induction, using this for the base case and Lemma 2 for the inductive step, this implies that $\forall N' \geq N\ \forall b' \in chain(b)\ p_{b'}^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$. Then, given any slot $N' > N$, we have $\forall b' \in chain(b)\ p_{b'}^{N'-1} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$, which by Lemma 1 implies that b is canonical in honest views of voters at slot N'. □

Algorithm 1 LMD Confirmation rule

```
State
    C : chain of blocks b
    b_0 : genesis block, i.e., block at slot 0 in C
    W_b^N ← |W_b^N|
    S_b^N ← |{v_i ∈ W_b^N : v_i voted for b}|
function parent(b)
    return b.parent
function isOneSafe(b, N)
    return S_b^N / W_b^N > 1/2 (1 + W_p / W_b^N) + β
function isLMDConfirmed(b, N)
    if b = b_0 then
        return TRUE
    return isOneSafe(b, N) ∧ isLMDConfirmed(parent(b), N)
```

3 Confirmation rule for the Ethereum consensus protocol

3.1 FFG filtering

The only blocks which are considered by the fork-choice are those in the subtree rooted at the latest justified checkpoint, *i.e.*, `store.justified_checkpoint`. Before running LMD-GHOST on that subtree, further filtering happens, potentially removing some branches. In the following, the filtering condition `correct_justified` is defined. A branch is not viable if `correct_justified` is false for the leaf block of the branch, and a block remains in the block tree (it is not filtered out) if it is contained in at least one viable branch.

```
correct_justified = (
    store.justified_checkpoint.epoch == GENESIS_EPOCH
    or voting_source.epoch == store.justified_checkpoint.epoch
    or (
        store.unrealized_justifications[block_root].epoch >= store.
            justified_checkpoint.epoch and
        voting_source.epoch + 2 >= current_epoch
    )
)
```

3.2 Withholding attacks

Let the current epoch be e, with epoch boundary block C. Suppose the adversary controls many slots in a row at the end of epoch e, enough that withholding those blocks will prevent the chain from containing sufficiently many votes to justify C. The adversary also controls the first slot in epoch $e+1$. Suppose also that it is able to use those blocks to create a fork which contains sufficiently many attestations to justify C, so that it will be justified in the state of the first block of epoch $e+1$, built on the adversarial fork. If that's the case, then the adversarial fork will have C as `state.current_justified_checkpoint`, and honest validators will set `store.justified_checkpoint` to C. On the other end, `store.unrealized_justifications[block_root].epoch` will be less than e for the leaf block on the honest branch, since it does not contain sufficient justification evidence. Therefore, `correct_justified` will be false for it, filtering out all blocks in the honest branch which are not also part of the adversarial branch, causing a reorg to the latter.

This is easiest if the fork is "shallow", *i.e.*, if it starts only a few blocks before the first of the consecutive adversarial slots, because the attestations in all blocks which are not forked out contribute to justification already. Any attestations in blocks which are forked out can still be used, but have to be included in some block on the adversarial branch. The adversary has to then ensure it has enough space to include attestations with total weight $\geq \frac{2}{3}W_t$, so

that C is justified on its branch. Even such shallow forks can break the safety guarantees of our confirmation rule, but it is also worth nothing that deeper forks are possible if the adversary controls sufficiently many slots. For example, suppose the adversary controls the last 11 slots in epoch e and the first one in epoch $e+1$. It can then create a fork which starts directly after block C, and continues with blocks from the 11 consecutive adversarial slots. Since $11 > \frac{32}{3}$, the honest fork cannot possibly contain enough attestations to justify C, as it only contains at most the first $21 < \frac{2}{3} * 32$ blocks of epoch e. Moreover, the adversarial branch can include enough attestations to justify C, since the redundancy in the space for including attestations in blocks is $2x$. The reorg described in the previous paragraph can then be carried out, which in this case forks out the entire honest branch after C.

3.3 Fork-choice change

Ensuring against this attack requires either already observing sufficiently many FFG votes on the current chain, in order for a justification to happen after the epoch transition, or to add additional assumptions, under which we have high confidence that enough votes will accrue in the honest chain by the end of the epoch. The first approach can work when confirmation is performed later in an epoch, because in practice sufficiently many votes will be observed soon after $\frac{2}{3}$ of the epoch has passed, at which point safety is ensured. For the earlier blocks, we are left with the second approach, but this is not very promising. In order for the attack not to be viable, we are forced to assume a fairly low β, both so that the adversary cannot withhold too many votes and so that it has a negligible probability of controlling many slots in a row at the end of the epoch.

To avoid having to go down this route, greatly weakening the confirmation rule, we can introduce a simple fork-choice change, which implements a strictly weaker form of filtering: at epoch e, with latest justified checkpoint J, a validator considers a leaf block b viable if it descends from J *and* the latest justified checkpoint in the state of b is *either* J *or* is from epoch $\geq e-2$. In other words, `correct_justified` is changed to the following, removing the condition on unrealized justifications:

```
correct_justified = (
    store.justified_checkpoint.epoch == GENESIS_EPOCH
    or voting_source.epoch == store.justified_checkpoint.epoch
    or voting_source.epoch + 2 >= current_epoch
)
```

This gets us closer to the "ideal" protocol with no filtering, where the whole subtree rooted at the latest justified is viable. In particular, as we explain in the following section, it prevents the withholding attack explained above, by affording us one extra epoch to include justification evidence, making it much harder for the adversary to prevent its inclusion and force filtering of an honest branch.

3.4 Analysis of the fork-choice change

No surround voting Consider an honest validator voting at epoch e, following the filtering rule specified in the previous section. If `store.justified_checkpoint.epoch == GENESIS_EPOCH` or `voting_source.epoch == store.justified_checkpoint.epoch`, then clearly the validator cannot possibly be committing a slashable offense with its vote. If `voting_source.epoch + 2 >= current_epoch`, then the vote is either from epoch $e-2$ to e or from $e-1$ to e. In either case, it cannot be a surround vote, since there is at most one epoch in between source epoch and target epoch, not enough to surround another vote.

Withholding attack prevention Consider the setup of the withholding attack above, where the adversary has managed to create an adversarial branch which justifies C, the EBB of epoch e, while on the other end the honest branch does not even include sufficient justification evidence for C. Still, the honest branch does *not* get filtered out by the new rule, *as long as it justifies epoch* $e-1$, because then `voting_source.epoch + 2 >= current_epoch` and `correct_justified` is true. At this point, the honest branch has the entirety of epoch $e+1$ to include sufficient justification evidence for C, and if it does so it will also not be filtered out in epoch $e+2$ either. For the attack to succeed, the adversary has two possible avenues: it either must prevent epoch $e-1$ from being justified, or it must prevent the justification evidence for epoch e from being included in the honest branch by the end of epoch $e+1$.

General prevention of filtering reorgs Say that we are at epoch e, and let the currently canonical head be B. We want to analyze the possibility of B being reorged due to being filtered out, because of `correct_justified` not being true at some point during epoch e. Without loss of generality, assume that epoch $e-1$ is not justified in the state of B, because otherwise `correct_justified` would certainly always be true for B throughout epoch e. We assume that epoch $e-2$ is justified, with justified checkpoint J. This amounts to two assumptions:

- The adversary cannot prevent the formation of *honest justification evidence* in an epoch, meaning that there are at least $\frac{2}{3}W_t$ honest votes from epoch $e-2$ (more generally, in any epoch) *with the same target*; and
- The adversary cannot prevent honest justification evidence from being included *in some chain* for a whole epoch: if honest justification evidence exists for epoch $e-2$, then it is all available by the end of the epoch, and we assume the adversary cannot prevent that, by the end of epoch $e-1$, there exists a chain which contains such evidence.

The first assumption is unavoidable, unless we do away with filtering altogether, and just require that a branch contains the highest justified checkpoint

in order to be viable [1](which requires far deeper protocol changes). This is because an adversary which can reliably prevent justification can prevent the honest branch from making any justification progress, and eventually reveal a branch with a higher justification, which causes a reorg. The second assumption can be made very weak, by either allowing blocks to include more attestations or by only doing so when they include sufficient evidence in order to justify. In particular, this second solution would mean that a single honest proposer per epoch is sufficient to satisfy the assumption.

Given that a justified checkpoint J from epoch $e-2$ exists, then it must be the case that J is justified in the state of B. This is because epoch $e-1$ is not justified in the state of B, so if it also did not justify epoch $e-2$ it would be filtered out, and not be canonical. Suppose now that B stops being canonical, while still being an LMD winner, due to being filtered out of the fork-choice tree. For that to be the case, it must be that J is not the highest justified. Therefore, there must be another justified checkpoint J' in epoch $e-1$. Let A be the block of J'. Since epoch $e-2$ is justified in the state of B, it must be the case that B conflicts with A, or it would not be filtered out. Let B' be the earliest ancestor of B which conflicts with A. Since J' is justified, there are $\frac{2}{3}W_t$ votes from epoch $e-1$ with target J', which must all be for descendants of A. At least $(\frac{2}{3}-\beta)W_t$ of these are votes from honest validators, and therefore they were cast timely and were public already at the end of epoch $e-1$. If αW_t is the total amount of weight which the adversary is willing to equivocate with, then the subtree rooted at B' can have received at most $(\frac{1}{3}+\alpha)W_t$ weight during epoch $e-1$, since all votes for A do not contribute weight to B'. Therefore, if $\frac{2}{3}-\beta > \frac{1}{3}+\alpha$, or in other words $\beta+\alpha < \frac{1}{3}$, it is the case that A must beat B' in LMD-GHOST, at the end of epoch $e-1$. Honest validators would then vote on the subtree of A, and it would only increase its LMD-GHOST advantage over the subtree of B', contradicting that B was canonical at some point in epoch e.

3.5 Extending the confirmation rule

Let:

- b be the block for which we seek confirmation;
- e be the epoch of b, *with* $epoch(N) = e$;
- $C(b)$ be the checkpoint in $chain(b)$ at epoch e, *i.e.*, the highest checkpoint in $chain(b)$;

[1]If we merely required being a descendant of the latest justified in order for a block to be viable, it could not be filtered out just because of the on-chain justification status. To filter out a block which would otherwise be an LMD-winner, the adversary would have to produce a conflicting justification, which is much harder. In fact, with the two assumptions we have made, the adversary is also forced to produce a conflicting justification, and in the next paragraph we show that this is very difficult

- $J(b)$ be the highest justified checkpoint in $chain(b)$. As part of the confirmation rule, we check that $epoch(J(b)) = e - 1$;
- W_f be the weight of validators yet to vote in epoch e *at the current time*;
- W_t be the total weight of the validator set; and
- let $S^{ffg,N}_{C(b)}$ be the FFG support of $C(b)$, *i.e.*, the weight from the subset of FFG votes received with $(\text{source} = J(b), \text{target} = C(b))$.

Assumptions:

- $a_1 : \ \forall N \geq n,\ A^N_b \leq \beta W^N_b < \frac{1}{3} W^N_b$;
- a_2: The adversary controls a fraction of validators $< \beta < \frac{1}{3}$;
- a_3: Suppose sufficiently many votes to justify checkpoint C are available at the end of $epoch(C)$, and that some block $b \succ C$ is canonical in all honest views at $epoch(C) + 1$. Then, by the end of epoch $epoch(C) + 1$, there exists some block $b' \succ b$ which contains enough attestations to justify C; and
- a_4: The adversary is willing to equivocate at most αW_t weight, with $\alpha \leq \beta$.

As we have discussed in the last paragraph of Section 3.4, a_3 can be made a much weaker assumption, if this ever were to be needed, by making small changes to the way attestations are included, so that a single honest proposer per epoch is sufficient.

Algorithm 2 Complete confirmation rule

```
1: function isConfirmed(b, N)
2:     bestDesc ← arg max_{b' ⪰ b ∧ epoch(b') = epoch(N)} S^N_{b'}
3:     return isConfirmedCurrentEpoch(bestDec, N)
4: function isConfirmedCurrentEpoch(b, N)
5:     return
6:         ∧ isLMDConfirmed(b)                                              ▷ (c1)
7:         ∧ S^{ffg}_{C(b)} − min(αW_t, β(W_t − W_f), S^{ffg}_{C(b)}) + (1 − β)W_f ≥ (2/3)W_t   ▷ (c2)
8:         ∧ epoch(J(b)) = epoch(N) − 1                                     ▷ (c3)
```

3.6 Proof

Firstly, we prove some lemmas relating to the basic induction of the vanilla confirmation rule which does not take into account FFG. To do so, we add the condition that b does not get filtered out, so that only LMD-GHOST votes are relevant to whether or not it is canonical.

Lemma 5. *If $p_{b'}^N > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$ and b is not filtered out by any honest validator in slot $N+1$, then all honest validators in slot $N+1$ will vote for a descendant of block b.*

Proof. We simple apply the reasoning from Lemma 1, together with the fact that b does not get filtered out. □

Lemma 6. *If $p_{b'}^N > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$ and b is not filtered out by any honest validator in slot $N+1$, then $p_{b'}^{N+1} \geq p_{b'}^N$ $\forall b' \in chain(b)$.*

Proof. We apply the same reasoning from 2, together with the fact that b does not get filtered out. □

We now show another lemma, which carries out the induction on p_b^N within a given epoch, again *provided that b never gets filtered out during it* (by any honest validator).

Lemma 7. *Consider an epoch $e' \geq e$, and let $N' = N$ if $e' = e$, or N' be the last slot of epoch $e'-1$ otherwise. If it is the case that b is never filtered out in the view of any honest validator during epoch e', and $p_{b'}^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$, then $p_{b'}^{N''} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$ holds $\forall N'' > N'$ with $epoch(N'') = e'$, as well as for N'. Moreover, all honest validators in such slots vote for a descendant of b.*

Proof. The induction is on N'', with the base case being N'. Let $N'' \geq N'$ be such that $epoch(N''+1) = e'$. We know that either $N'' = N'$ or $epoch(N'') = e'$. In either case, by the inductive assumption we have that $p_{b'}^{N''} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$. If b does not get filtered out by any honest validator during epoch e', then, by Lemma 6, $p_{b'}^{N''+1} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in chain(b)$. □

Lemma 8. *Suppose that every honest validator in an epoch e' votes for a descendant of block b. Then, no checkpoint conflicting with b can be justified in epoch e'*

Proof. Since all honest validators in epoch e' vote for a descendant of b, any branch conflicting with b can at most receive the adversarial FFG votes from epoch e'. By a_2, the weight from adversarial votes is $< \beta W_t < \frac{W_t}{3}$, so no checkpoint conflicting with b can be justified. □

Main Proof. In order to show that b is safe, we can show that any honest validator always sees it as canonical in every slot $> N$. For that to be the case, it is sufficient that, $\forall N' \geq N$, b is never filtered out of the block tree of any honest validator in slot $N'+1$, and that $p_b^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ holds, because then Lemma 5 implies that all honest validators in slot $N'+1$ see b as canonical.

To show this, we proceed by induction on the epoch. For each $e' \geq e$, we want to show that b does not get filtered out during epoch e, and also

that, for any $N' \geq N$ such that $epoch(N') = e'$, $p_{b'}^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ holds. Moreover, we also include in the inductive assumptions that there is no justified checkpoint of epoch $e'' \in [e-1, e']$ which conflicts with b.

The base cases are e and $e+1$. Let's first look at epoch e. Firstly, we show that b does not get filtered out at any point in epoch e. By c_3 (Line 8), $epoch(J(b)) = e-1$, and, by a_2, there cannot be conflicting justified checkpoints at any epoch, so $J(b)$ must be the highest justified checkpoint for any honest validator in epoch e whose view includes b. Since $J(b)$ is by definition justified in b, b is not filtered out by any such honest validators in epoch e.

By Lemma 3, c_1 (Line 6) implies $p_{b'}^{N} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall b' \in \text{chain}(b)$. Therefore, we can use Lemma 7, and conclude that $p_b^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall N' \geq N$ with $epoch(N') = e$. This in particular implies that all honest votes in the remaining slots in epoch e go to the subtree of b. Therefore, out of the remaining W_f votes from epoch e, $\geq (1-\beta)W_f$ will vote for $(source = J(b), target = C(b))$, since they vote for the subtree of b. By definition, $S_{C(b)}^{ffg}$ is the FFG weight which is supporting $C(b)$ already. Due to assumptions a_1 and a_4, by the end of epoch e, the maximum FFG weight that can be subtracted from $S_{C(b)}^{ffg}$ due to the adversary committing slashable offences is $\min(\alpha W_t, \beta(W_t - W_f), S_{C(b)}^{ffg})$, where $(W_t - W_f)$ corresponds to the weight of the validators that have yet to vote in the current epoch. Therefore, at the end of epoch e, $C(b)$ will have a total FFG support of $\geq S_{C(b)}^{ffg} - \min(\alpha W_t, \beta(W_t - W_f), S_{C(b)}^{ffg}) + (1-\beta)W_f$, and this is $\geq \frac{2}{3}W_t$ by c_2 (Line 7), so enough to justify it. By a_2, this implies that no conflicting checkpoint can be justified in epoch e, so that we have checked all conditions for epoch e.

Let's now move to epoch $e+1$. Firstly, we show that b does not get filtered out during it. For this to be the case, it is sufficient that $epoch(J(b)) =\geq (e+1) - 2 = e - 1$, which is true by c_3 (Line 8), and that b descends from the latest justified. Say the latest justified is J'. By c_3 (Line 8) and a_2, either $J' = J(b)$ or $epoch(J') = e$. In the first case, b clearly descends from J'. In the second case, $J' = C(b)$, because we have already shown that no checkpoint conflicting with b could be justified in epoch e. Therefore, b does indeed not get filtered out in epoch $e+1$. Using this, and that $p_b^{N'} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W_b^N})$ for N' the last slot of epoch e, we can now apply Lemma 7 to epoch $e+1$, and conclude that $p_b^{N'} > \frac{1}{2(1-\beta)}(1 + \frac{W_p}{W_b^N})$ $\forall N'$ in epoch $e+1$. In particular, this implies that b is canonical in all honest views in epoch $e+1$, and thus that the subtree of b receives all honest votes. This immediately implies that no checkpoint conflicting with b can be justified in epoch $e+1$, by Lemma 8.

Before moving to the inductive step, for epochs $> e+1$, we need one last piece, *i.e.*, that $C(b)$ is justified in *some* branch descending from b, after the

epoch transition to $e+2$. To do so, we apply a_3, which we can do because we have shown that enough votes to justify $C(b)$ are publicly available by the end of epoch e, and that $b \succ C(b)$ is canonical in all honest views in epoch $e+1$. We conclude that, by the end of epoch $e+1$, there exists a block $b' \succ b$ which contains enough attestations to justify $C(b)$, so that in fact $C(b)$ is justified in this branch after the epoch transition to $e+2$.

Finally, we can move to the inductive step. Consider some epoch $e' \geq e+1$, for which all inductive assumptions are satisfied. Firstly, we show that b does not get filtered out in epoch $e'+1$. At some point in epoch $e'+1$, let J' be the highest justified. If $J' = C(b)$, then b does not get filtered out, because $C(b)$ is justified in $b' \succ b$ from the beginning of epoch $e+2 \leq e'+1$, so the branch of b' does not get filtered out. If $b \prec J'$, then clearly it does not get filtered out, because the filtered tree will always contain the justified checkpoint, even if there are no viable leaves. Therefore, b can only be filtered out if J' is conflicting with b. By inductive assumption, there is no justified checkpoint in epochs $[e-1, e']$ which conflicts with b, so J' cannot conflict with b, since $epoch(J') \geq epoch(C(b)) = e$. Thus, b does not get filtered out.

As we have done in epoch $e+1$, we use this, and that $p_b^{N'} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W_b^N})$ for N' the last slot of epoch e', to apply Lemma 7 to epoch $e'+1$, and conclude that $p_b^{N'} > \frac{1}{2(1-\beta)}(1+\frac{W_p}{W_b^N})$ $\forall N'$ in epoch $e'+1$. In particular, this implies that b is canonical in all honest views in epoch $e'+1$, and thus that the subtree of b receives all honest votes. As before, this immediately implies that no checkpoint conflicting with b can be justified in epoch $e'+1$, by Lemma 8.

□

4 Probability calculation on the assumption of ratio of adversarial LMD votes

In this section we compute the probability that the assumption $\forall N \geq n$, $A_b^N \leq \beta W_b^N$ holds.

Note that this section is still incomplete.

4.1 Ratio of adversary votes for two slots across an epoch boundary

The first scenario consists in two consecutive slots across an epoch boundary. In particular, we are interested in the probability that up to the end of the second slot, the percentage of adversary votes to the total votes is at most β. Note that some votes in the first epoch may be overlapped by votes in the second slot cast by same set of validators.

We calculate it by computing

$$\sum_{\frac{i}{j} \leq \beta \wedge i \geq 0 \wedge j \geq S} Pr(A = i \wedge W = j) \tag{1}$$

In Eq. 4.1, i denotes the weight of effective adversary votes in the two consecutive slots, and j denotes the weight of total effective votes in the two slots, where the first slot (slot 0, with S_0 the set of votes from its committee) is the last slot in the former epoch up to the epoch boundary and the second slot (slot 1, with S_1 the set of votes from its committee) is the first slot in the latter epoch starting from the epoch boundary. Let C_0 be the total adversary votes and C_1 be the total honest votes, with $w = |C_0| + |C_1|$ and $C_0 \cap C_1 = \emptyset$. Equivalently, we may (informally) write $C_0 = \beta \cdot w$. The voting weight of each slot is $S = |S_0| = |S_1|$. $Pr(W = j)$ is then the probability that the "effective" weight of total votes in the two slots is j, *i.e.*, $W = |S_0 \cup S_1| = j$, and likewise, $Pr(A = i \mid W = j)$ is the (conditional) probability that given total "effective" weight j the total "effective" adversary weight is i.

For computing $Pr(A = i \wedge W = j)$, there are i effective adversary votes and $j - i$ effective honest votes to be taken into account altogether, and $i \leq j$. In order to enumerate all possible ways of getting i adversary votes in the union of two slots, (1) we first select k adversary votes for S_0, before completing the remaining with $S - k$ honest votes, and we require both $0 \leq k \leq i$ and $0 \leq S-k \leq j-i$. Then (2) for S_1, we select $i-k$ non-overlapping adversary votes out of C_0-k adversary votes non-overlapping with S_0, followed by $(j-i)-(S-k)$ honest votes out of $C_1 - (S-k)$ honest votes non-overlapping with S_0. Finally, (3) we complete the remaining (overlapping) votes in S_1 out of S already selected votes. We split in the following cases.

(1) $\binom{C_0}{k} \cdot \binom{C_1}{S-k}$

(2) $\binom{C_0 - k}{i - k} \cdot \binom{C_1 - (S-k)}{(j-i) - (S-k)}$

(3) $\binom{S}{S - (i-k) - ((j-i) - (S-k))}$
which is to select what remains in slot S_1 of size S after whatever are selected in (2)

After simplification for the cases in (1)-(3), we have $Pr(A = i \wedge W = j)$ with the following value. The sample space (the denominator) is formed by selecting two slots of votes with replacement.

$$\sum_{0 \le k \le i \wedge 0 \le S-k \le j-i} \frac{\binom{C_0}{k} \cdot \binom{C_1}{S-k} \cdot \binom{C_0 - k}{i-k} \cdot \binom{C_1 - S + k}{j - i - S + k} \cdot \binom{S}{2S - j}}{\binom{w}{S} \cdot \binom{w}{S}}$$

4.2 A general formula for the ratio of adversary votes by the end of the next (new) slot

This section is yet to be completed.