1 Reward-Penalty Model

Let $V \in \mathcal{V}$ be a validator. Let D_i^V be the deposit (in `ether`) of V in epoch i. Let $\mathcal{D}_i \stackrel{\texttt{def}}{=} \Sigma_{V \in \mathcal{V}} D_i^V$ be the total amount of deposits (in `ether`) in epoch i. Let R_i be a reward-penalty factor in epoch i.

In each epoch i, validator V gets reward and/or penalty as follows.

- V pays a fee for each epoch whether or not he votes.
- V gets a reward, if he votes "correctly" — the source epoch of the vote is equal to the recommended one.
- V gets another reward, if epoch $i-1$ is finalized at the end of epoch i.

Given D_i^V and R_i, D_{i+1}^V is defined for each case as follows. Note that "incorrect vote" entails "no vote".

$$D_{i+1}^V \stackrel{\texttt{def}}{=} \begin{cases} D_i^V \cdot \dfrac{1}{1+R_i} & \text{if } V \text{ does } \textit{not} \text{ vote (correctly), and epoch } i-1 \text{ is } \textit{not} \text{ finalized} \\ D_i^V \cdot \dfrac{1}{1+R_i} \cdot (1 + \dfrac{\alpha}{2} R_i) & \text{if } V \text{ does } \textit{not} \text{ vote (correctly), but epoch } i-1 \text{ is finalized} \\ D_i^V \cdot \dfrac{1}{1+R_i} \cdot (1 + R_i) & \text{if } V \text{ votes correctly, but epoch } i-1 \text{ is } \textit{not} \text{ finalized} \\ D_i^V \cdot \dfrac{1}{1+R_i} \cdot (1 + R_i) \cdot (1 + \dfrac{\alpha}{2} R_i) & \text{if } V \text{ votes correctly, and epoch } i-1 \text{ is finalized} \end{cases}$$

Here, α is the fraction of the correct votes in the total deposit $\mathcal{D}_i$. Since α is used only when epoch $i-1$ is finalized (which implies the current epoch i is justified), we have $\frac{2}{3} \leq \alpha \leq 1$.

At the beginning of the next epoch $i+1$, the reward factor R_{i+1} is also adjusted based on the current total deposit and the history of finalization as follows:

$$R_{i+1} \stackrel{\texttt{def}}{=} \frac{\beta}{\sqrt{\mathcal{D}_i}} + \gamma \cdot (\mathsf{ESF} - 2)$$

where β is a fixed base interest factor, and γ is a fixed base penalty factor. ESF is the number of epochs since the last finalized epoch. We have $\mathsf{ESF} \geq 2$, at the beginning of epoch $i+1$, since the latest possible finalized epoch is $i-1$.

Lemma 1. *We have the followings:*

- *If V votes correctly, his deposit* never *decrease, i.e., $D_{i+1}^V \geq D_i^V$.*
- *If V does not votes correctly (or does not vote at all), his deposit strictly decreases, i.e., $D_{i+1}^V < D_i^V$.*
- *In an ideal situation (all validators vote correctly and every epoch is finalized), each validator's deposit strictly increases for each epoch, i.e., $D_{i+1}^V > D_i^V$, and the reward factor strictly decreases for each epoch, i.e., $R_{i+1} < R_i$.*
- *The above holds for both positive and negative γ.*

Relationship to the contract source code In the `initialize_epoch` function[1]:

- $D_{i+1}^V \times 10^{18} \simeq$ `self.validators[`V`].deposit` $\times$ `self.deposit_scale_factor[epoch]` at line 273.
- $R_{i+1} \simeq$ `self.reward_factor` at line 284.

[1] `https://github.com/ethereum/casper/blob/b2a1189506710c37bbdbbf3dc79ff383dbe13875/casper/contracts/simple_casper.v.py`

- $i + 1 =$ `epoch` at line 266.
- $\sqrt{\mathcal{D}_i} \simeq$ `self.sqrt_of_total_deposits()` at line 276.
- $\mathsf{ESF} =$ `self.esp()` at line 277.
- $\alpha \simeq$ `vote_frac` at line 231 of the `collective_reward` function (called at line 270).
- $\beta =$ `self.BASE_INTEREST_FACTOR`
- $\gamma =$ `self.BASE_PENALTY_FACTOR`