My understanding from the paper:

- PoS validators determine the most valid chain at every 100th block
- validators vote on PoW blocks, they don't generate their own
- tip after a most recent checkpoint is PoW based
- for PoW to exist there has to be a block reward

Therefore:

1.) Hostile PoW attack.

In the case of a hostile PoW attack, validators are likely to cooperate and ignore the hostile PoW chain to save the network. It only requires a relatively short period of time to bring the PoW difficulty down. After that, validators start to mine low-difficulty PoW blocks - these blocks are initially ignored by other nodes. Then they create a checkpoint, making it the most valid chain. The rest of the network forks to their chain.

2.) Profit-maximizing validators.

Validators have a strong incentive (PoW block rewards) to conspire and ignore external PoW even in the absence of an attack. Not a realistic concern due to being invested for the long-term.

In both cases ethereum functionally changes to one 'megablock' (divided into 100 blocks) per ~23 minutes.

What (1) means is that PoW is irrelevant to the security, what (2) means is that validators are (implicitly) assumed to not be "rational" but at least a bit friendly, which makes PoW pointless in itself.

I propose a very simple alternative to PoW in FFG

. It's not meant to be the ideal end solution, merely better than PoW while being very simple to implement. Global PoW waste is infuriating.

I

- target PoS inflation (interest) per epoch, in percents, assuming infinite coins vote

coins

- how many coins in total

V_e

- ordered set (array) of active validators (accounts) for epoch e

V_e.stake

- how many coins these validators have in total

I_e

- effective inflation rate for epoch e

F_e

- total gas fees for epoch e

(a) for every epoch e

scale the interest rate according to the logistic function, ie:

I_e

= I

- sigmoid(k
- V_e.stake

/coins

)

Where k

is the scaling factor.

At every checkpoint that ends epoch e

:

(b) Scale the deposit for all active validators:

for v

in V_e

: v.stake

*= (1 + I_e

)

(c) divide all fees from the epoch proportionally amongst all active validators according to their deposits:

for v

in V_e

: v.stake

+= F_e

- v.stake

/V_e.stake

Which allows a simple hash-based block generation algorithm:

Given most recent block B

, its hash H(B

), current unix timestamp t

and target block time z

(seconds), validator i

generates a child block when:

H(B

. t

) % (z

*|V_e

|) == i

The more valid short pseudo-PoW chain is simply the one with more blocks, or if equivalent, the one that was seen first by the node.

Intended effect of incentives:

1. No validator cabals under condition:

Due to the sigmoid scaling of a block reward in (a), existing validators happily include new validators, as long as profit from a higher interest rate is higher than loss due to fee sharing. That’s why the interest rate must be strictly monotonically increasing.

Finding the ideal scale factor k

requires currently untestable assumptions about future proportion of voting coins and average fees during epoch.

With k

= 3 sigmoid rises from 0.5 at 0 to 0.817 at 0.5 - or equivalently 1% and 1.63% for 2% rate in the limit - which seems reasonable as an initial value.

1. No stake grinding and similar concerns, as fees are distributed among all validators.

2. Strictly monotonically increasing interest rate creates hard to quantify positive social effects - as every staking newcomer is directly

beneficial to all others. Constant interest rate - or even, worse, decreasing - makes every newcomer a loss, in addition to perverse incentives due to fees.

Possible problem:

Free riding by not generating individual blocks. It’s another consequence of the ‘profit-maximizing validators’ assumption which imo isn’t realistic. In any case, it’s a much milder outcome than possible PoW hijacking.