In the Casper testnet interest rate is proportional to inverse square root of total deposits. That would be very bad for full PoS.

Changes in validator sets are transactions.

Bob is a selfish profit-optimizing validator. If Bob's expected future return on staked eth is expected to drop due to new validators joining he's not going to include their join transactions. Fee loss is infinitesimal under any significant adoption scenario, as he only loses the difference to the next highest gas priced transaction.

The simplest case is:

Bob is the last block generator in a period (anything from a block to epoch) that's used to count total deposits and interest rates.

If it's per epoch and Bob is making the last block he has a 100% chance to block new validator for at least one epoch.

If it's per block then for at least one block.

In general Bob should censor when

lostFeeDifference < profitlfNotIncluded*probabilityOfNotInclusion (by the remaining block producers, if any)

The BAD thing is this censorship automatically works as signalling. If Bob observes other validators behaving the same way it changes the probability of not inclusion.

It can catastrophically end up in a state in which no new validators are allowed to join due to censorship, forming a cartel than owns the network.

A nice theoretical solution would be to make validator joins zero-knowledge, provided that indistinguishable zk transactions are common on the network, as that would make censorship impossible. Currently not realistic.

In the absence of that, the individual profit incentive has to be to include new validators, which means that interest rate must rise with the total staked amount. It can't be constant because fees would effectively make it decreasing. Ideally it would depend on fees but that's too gameable.

I propose a nice sigmoid:

annualRate = rateScaling * sigmoid(k * fractionOfStakedEth)

With k = 3 and rateScaling = 2.1% it goes from 1.05% rate at 0 eth staked to 1.7% rate at 50% staked; ~2% rate at 100% eth staked.