Hello, community,

I was thinking about nothing-at-stake problem in PoS, and why actually we need to have slashing conditions. Let consider PoS algorithm which uses a stake to solve Sybil attack issue and after that just runs BFT agreement protocol on 3t+1 validators. If adversary controls up to t committee members - everything is fine. However, if the adversary controls t+1 committee members out of 3t+1 and achieves some significant control over the network, he can achieve a fork. Let me briefly describe this attack.

Let A and B be two disjoint sets of honest nodes, but adversary controls messages delivered to them (network partition stage) and C - set of corrupted committee members. |A| = |B| = t = |C|-1. Now let adversary finalize the block in partition A+C, and assure that anything was not sent to B about that block. B starts network recovery mode and recovers with set C. After that new block is finalized in the partition B+C, distinct from block finalized in partition A+C. Now, C controls two different chains. Hence, a double spend is possible.

However, in Casper consensus, such is not possible. The adversary must control more than 2/3 of the total stake to perform double spend attack, and if he is somewhere in the range of 1/3 to 2/3, an attacker can only successfully halt the addition of new blocks to the chain, and not for a long time due to inactivity leak slashing conditions. Of course, this claim needs to be proven, but the attack described above is not possible with Casper since the attacker must sign two different blocks on the same height and that is immediately penalized by slashing conditions on any chain after the split.

Is that the main difference of Casper and BFT algorithms? Casper will properly function even if an attacker has up to 2/3 of the stake?