Summary Alternative Consensus 1

Tim Sigl

The Avalanche network uses a leaderless protocol. Meaning there is not a single last block like in Bitcoin where the next block is appended to. Another feature is that the knowledge of all nodes is not a requirement for the network to operate. When the number of malicious nodes increases the network stability degrades gracefully. The proposed consensus family ‘Snow’ tries to combine the best features of classical consensus protocols, where final consensus is reached after each consensus round and Nakamoto style consensus mechanisms, that are more scalable. Snow has a probabilistic safety guarantee, no PoW, different parameterization options to switch from more safetines to more liveness and uses a directed acyclic graph as the underlying data structure. The protocol is layered into 3 different sub-protocols. The first one Slush is not even Byzantine fault tolerant. This tolerance is built by the next layers Snowflake and Snowball. Slush communicates with its neighbors and switches its state depending on the majority opinion. The next layer Snowflake captures how often the same state was chosen. Finally, the upper most layer Snowball adds a confidence counter that switches to another state just when it is more confident in the new state. There is also a peer-to-peer payment system built on top of the proposed consensus family. The issue of double spending is solved by repeated runs of the Snowball algorithm to calculate the confidence. In experiments the creators were able to achieve 7000 tps with 125 nodes. They noticed only a small degradation in tps with more nodes. Also the network has a low latency due to the small block size.

Algorand uses a different approach to consensus via Byzantine Agreements. BA reaches consensus among all its users on the next set of transactions. There are two different roles in the consensus. Users can be selected as committee members to agree on proposed blocks or as block proposers to create new blocks. Users do not keep any state, so they can be easily switched out for different users in the consensus process. Users calculate on their own depending on a public seed if they are chosen as a committee member or block proposer. Other users do not know who got selected to which group to prevent eclipse attacks. Forks in Algorand are very rare. A 1 MB block can be confirmed in 22 seconds with 50.000 members. Users will gain voting power depending on the amount of tokens they hold. At least ⅔ of all tokens have to be held by honest nodes for the consensus protocol to work properly. There are different states of consensus that can be reached depending on how synchronous the network is. Messages are only gossiped to a random and small set of peers. Experiments have shown that the optimum number of block proposers is 26. Each block proposer has a different priority. The priority tells the committee which blocks to discard. The best time trade-off between block proposals is 5 seconds. In each round of voting each member casts a vote and also counts the vote of other members. If the network splits into two or more partitions a fork is proposed that all users should agree on. The latency in Algorand is consistent even if more users join the network.