Background: https://vitalik.ca/general/2018/08/07/99_fault_tolerant.html

If we want 95% fault tolerance, then in order to achieve a 2^{-40}

(~1 in 1 trillion) rate of failure, we need to have enough randomly sampled nodes that there is a 2^{-40}

chance that all of them are attackers, which requires log(2^{-40}) \div log(0.95) \approx 540

nodes. This implies that if we want to survive a network latency of \delta

, the the per-participant extension period would need to be 2 * \delta

, and so the entire algorithm would need to take 1080 * \delta

time to run. Given that the network latency assumption itself must be very conservative, this is highly suboptimal.

We can improve on this algorithm, getting fault tolerance 1 - \frac{O(1)}{n}

in n

rounds, as follows. Suppose that we pick a large set of nodes (approaching infinitely large), and arrange them all into subsets of n

, each of which runs the consensus in parallel. If the attacker controls a share of \le 1 - \frac{ln(2)}{n}

, there is a probability < \frac{1}{2}

that the attacker will fully control any given set. Each user can accept the output of the consensus as being the modal (ie. most frequent) result of the individual consensus processes. Hence, the attacker would need to corrupt more than half of the sets, and as the number of sets approaches infinity the probability of this approaches zero.

More concretely, suppose there are 700 sets running in parallel, and we target a fault tolerance of 1 - \frac{1}{n}

. There is a chance of \frac{1}{e}

that the attacker will fully control any given set. The probability that the attacker will get a majority is \approx 2^{-40.36}

. If we relax the fault tolerance to 1 - \frac{2}{n}

, so there is only a chance of \frac{1}{e^2}

that the attacker will fully control any given set, then only 68 sets will suffice, and with 1 - \frac{3}{n}

fault tolerance only 32 sets will suffice.