Incorrect by Construction - CBC Casper Isn't Live

Pyrofex Corporation

Many of the essential ideas in CBC Casper [1] are genuinely innovative, but the protocol is not live even under very strong assumptions about the network. I.e., even when the network is operating correctly, the protocol may not be able to achieve consensus on any blocks. It is therefore unsuitable for use in a blockchain.

Consider a synchronous, reliable network of four honest, equally weighted validators w_i for $0 \leq i \leq 3$, where any message sent must be received within two timeslots, and messages are never dropped, duplicated, or forged.

We begin with a genesis block G, on which w_0 and w_1 build blocks. Seeing w_0's block, w_2 builds on w_0. Likewise, seeing w_1's block, w_3 builds on w_1. However, before seeing w_2's block, w_0 sees w_1 and w_3's blocks and repudiates its first block, building on top of w_3. Symmetrically, w_1 sees w_0 and w_2's blocks before seeing w_3's block and repudiates its first block, building on top of w_2. Proceeding forward in this fashion, at each timeslot a validator repudiates a block and changes branches, resulting in the DAG in Figure 1. Adversaries needn't tamper heavily with the network to starve it. Indeed, even under normal operating conditions, a network using Casper could be extremely slow, or unable, to reach consensus.

This is troubling, since to starve the network we need not assume *any* Byzantine faults, nor extraordinary network conditions.

In fact, our demonstration assumes an ideal network. No real-world distributed system, much less a trustless blockchain, could implement a synchronous, reliable network of purely honest validators where no messages are lost, duplicated, or forged. Even under these pristine conditions, Casper fails.

Notice, even if CBC Casper is safe—to wit, no two nodes will ever decide on different values—it hardly matters, because the nodes may not decide at all. Others [2] have noted that safety and liveness are related requirements for blockchains. We concur.

As a result, we advise extreme caution when considering CBC Casper for use in any distributed systems, but particularly economically sensitive systems like a blockchain.

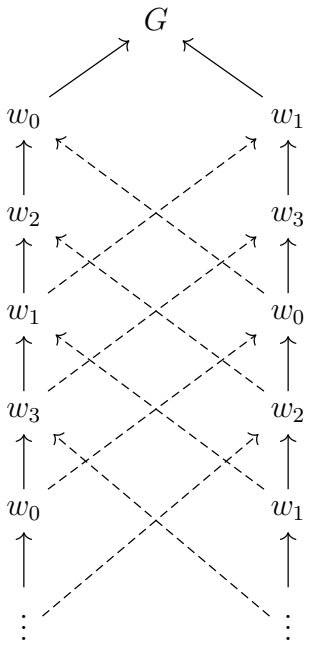

Figure 1: We can starve Casper by delaying message delivery one timeslot. Dashed lines indicate repudiation. The blocks are irrelevant, so we label vertices by the block's publisher.

References

[1] Zamfir, V., "A Template for Correct-By-Construction Consensus Protocols", available at https://github.com/ethereum/research/blob/master/papers/cbc-consensus/AbstractCBC.pdf.

[2] Ali, M., "Peer Review: CBC Casper", available at https://medium.com/@muneeb/peer-review-cbc-casper-30840a98c89a