Hi Everyone,

This is my first post, and hopefully more later

I think Casper (or other protocols with weak subjectivity) would benefit from formats for “the weak subjectivity information”, so that people can easily and safely share the information required to synchronize with the consensus:

1. a list of public crypto credentials and weights associated with the validators,
2. a block finalized by that validator set.

The above is required to make a fork-choice and to tell when new blocks are finalized. But we naturally would like to replace this with the following (which represents the “status quo proposal”):

1. a single block hash.

Armed with a single block hash, you could connect to the network and ask peers for (and hopefully receive):

1. a block with that hash
2. merkle proofs for the block’s current validator set
3. a block finalized by the block’s current validator set*

And then after receiving and authenticating the block’s hash, the merkle proofs, and finality on the other block, you can safely use the fork-choice rule.

This (or something like this) can be made to work, but we did make a couple of background assumptions:

1. you are able to peer
2. the weak subjectivity information is not expired

Clients are already required to have at least a hash in order to authenticate the consensus (fork choice and finality) for the first time, so I’d like to require some additional information:

[IP_address, block hash, expiry date**]

The IP address is trusted to provide peers who can provided information that can be authenticated via a hash collision with the block hash. The block hash is trusted information (trusted to have a currently-bonded validator set), and the expiry date is just there to prevent the user from using stale authentication information.

The main advantage of this proposal is that it removes the need for bootstrap seed nodes which provide clients with peers.

The challenge is to make it as easy and safe as possible for people to share this information. One idea is to encode the information in a QR Code, which is absolutely useful. However, for a human readable format (something that could be easily written down by hand and read aloud), the following might represent an improvement:

[domain name in DNS, first 14 characters of Base58Encoded(block hash), expiry date]

I chose “Base58” encoding (used in Bitcoin) because it’s a bit safer than base 64, but base 64 would provide more compression. Given the odds of a mistyping will still be the hash of an invalid block, I think of using base 64 as having a liveness bug for users who don’t enter the hash correctly.

I went with DNS instead of ENS because using ENS requires already having peers (just as DNS requires having a DNS server, but presumably that’s easier to do). All in all, I think the choice between using DNS and using an IP address is about whether a usability/security trade-off is worth the gained readability.

I think there’s probably lots of room for improvement and other people’s thoughts!

Vlad

*It may be reasonable to do this with a new request type, or to look into the state trie for the last finalized block. But then it must be the case that the hash that you received has a block with state that contains the hash of a block finalized by that validator set

** should be a human readable date, rather than a block number or something like that