Blockchain Abbreviation

MAXIM AMELCHENKO

SHLOMI DOLEV

TO APPEAR IN IEEE NCA2017

Blockchain

- A distributed ledger with no single trusted authority.
- All participants take part in verifying the data appended to the Blockchain, thus creating a decentralized trust mechanism.
- The blocks appended to the chain are cryptographically chained together (using hash functions), i.e. changing a content of a certain block will require a change in the entire suffix succeeding this block.
- The Blockchain is effectively immutable.
- Adding a new block requires a lot of computional power (a process called "mining").
- The entire network, possibly containing minority of Byzantine participants, enforced to keep working sequentially and constantly add more blocks

“Popular” rough description

- https://www.youtube.com/watch?v=lik9aaFIsl4

Blockchain Size

- Over time the continuously added blocks form a long chain which grows in size, 130GB for Bitcoin in October 2017.
- This size problem has already raised flags with a lot of Blockchain service providers, which offer lightweight clients. Those clients do not download the entire chain, but keep only a small fraction of it locally, while relying on third parties to deliver verified blocks.
- This is not an ideal setting, since how can you trust a third party to deliver you the right blocks?
- Enter Blockchain Abbreviation.

Blockchain Abbreviation

- After we've established a need to shorten the Blockchain, let's look at the possibilities we have. Naturally the protocol will need to change somehow to accommodate any deviation from standard procedure.
- We cannot cut parts of a Blockchain because we'd ruin the cryptographic connections of the remaining blocks.
- We cannot erase information from blocks. That would alter their content, thus changing the hash of the block, disrupting the connections between the blocks.
- How about replacing the entire Blockchain with a summary?

Ethereum

- A cryptocurrency similar to Bitcoin, with some cool additions:

- Smart Contracts.

- Block generation time is much smaller.

- The first block (called "Genesis") can preallocate funds to accounts.

We can use that last part, to replace the entire blockchain with a new Genesis block containing the balances of all accounts.

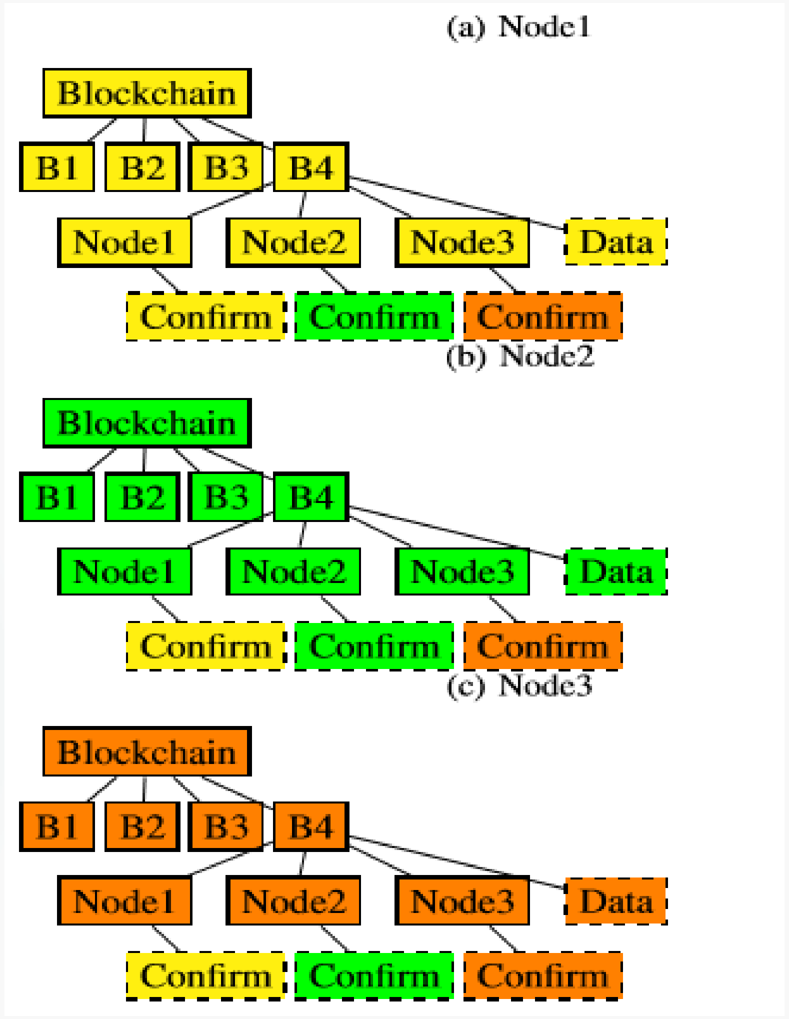
Ethereum protocol changes

- A change like this would not pass without protocol changes.
- Block size limit needs to change. With time, the number of accounts may be so big, that the new Genesis block will be big in size, still much smaller than the transaction it summarizes. The default block size limit might not be enough.
- Participants must know of this new feature and accept the abbreviated Blockchain.
- More changes might be needed for this feature to be fully integrated to the Ethereum network.
- We've implemented this scheme with the Ethereum project. By manually planting a new Genesis block into miners' data files, we were able to verify the miners continue to mine from the new state.

UNIX based Implementation

- Another option is to provide a new Blockchain system implementation which eliminates the problems we are having.
- We provide a UNIX based implementation of Blockchain. This system is not cryptocurrency based, but deals with records published in the ledger. There is no need for accounts or transaction in this type of Blockchain.
- Blocks are still “mined” in the traditional way, and linked using hash functions.
- In this implementation, we can actually cut any prefix of the chain, since we don’t need any account balances to maintain. So the Genesis block does not have any special information it needs to hold, so it can be any of the mined blocks which wasn’t cut.

UNIX based Implementation



UNIX based Implementation

- Our implementation is a permissioned ledger where all nodes know each other and connect to each other in constant intervals.
- We use SSH for connecting, and each node has a user account in all other machines. All of those users belong to the same group.
- Instead of sending each other blocks, nodes learn about new blocks when they connect to other nodes and after verification copy them over to their local chains.
- Abbreviation is implemented by voting. Each node writes a confirmation file for every block it inspects during connection to other nodes. This file is protected by UNIX permissions, so only a user can write its confirmation file. When it erases a confirmation file its a vote to abbreviate a chain. When a majority erases a confirmation file from a certain block, the next block becomes the new genesis block and all preceding blocks are erased.

Abbrev.