A Proof-of-Work Hybrid with Nxt

Alexander Dolgunin*

NXT-2B3E-AUU2-GFCP-EUT4S

a.dolgunin@gmail.com

Abstract

Taking NxtClone software repository as it's main codebase, a hybrid borrowing code from BitcoinJ Java implementation of Bitcoin protocol was created, with two major kinds of blocks: key blocks whose header resembles Bitcoin's with some additions and alterations; fast blocks that originate from Nxt blocks with some modifications to make them more compatible with key blocks under a single block concept. The Blockchain has stable MainNet since 8 Aug 2018 with thousands of key blocks interspersed with hundreds of fast blocks between each two key blocks. The project's history, motivations and future research opportunities are discussed.

Contents

I. Introduction

Consensus protocols allow the blockchain to confirm transactions without relying on a third party. They allow a decentralized network to arrive at an agreement about the state of it's blockchain, also called "the true state" or "truth". In the following section, we will discuss two of the most popular protocols: Proof-of-Work (originating in Bitcoin, Jan 2009) and Proof-of-Stake (Peercoin, Aug 2012 and Nxt, Nov 2013).

*Alexander was part of the original team who delivered Metro Blockchain

II. The Problem

i. Limitations of Pure Proof-of-Work

The first — and still most common — distributed blockchain protocol mechanism is called "Proof-of-Work." It's most closely associated with Bitcoin, which introduced it to the world, but many cryptocurrencies have subsequently adopted it. It's designed for a public (or permissionless) blockchain.

Despite the original decentralized vision of Bitcoin's inventor that went under pseudonym of Satoshi Nakamoto, in practice Proof-of-Work (called **PoW** for brevity) led to centralization due to introduction of ASICs and mining pools, which we will discuss in more detail in III, ii. Currently the profitability of mining is diminishing, due to insufficient growth of networks lacking real-life applications, with a possible solution such as:

- Dis-intermediate the pools by removing the need for them: distribute "advance payments" through Consensus to miners who found a partial hash (a hash short of one or two binary zeros to satisfy the target)
- Using PoS component, make mining ASIC and GPU-proof so that every CPU can mine
- Reward miners and stakers with emission for social activity leading to growth of the network

ii. The Distribution Problem of Pure Proof-of-Stake (PoS)

One disadvantage is that in pure Proof-of-Stake, the only way to acquire coins is from someone who already has them. This can lead to issues with the distribution. For example, in Nxt the entire coin supply was initially distributed to 73 "founders", and some of those people still own significant fractions of the supply, giving them wealth and influence. That said, none now own as large a fraction as Satoshi owns of Bitcoin. Although it may seem that the distribution problem will solve itself over time, as the

founders have an interest in spending their wealth to support the coin, the established Pure PoS systems still haven't reached their full potential.

iii. The Nothing-at-Stake (N@S) Vulnerability

In PoW miners who produce blocks need to decide which chain, sometimes called "version of reality", they support by building their blocks on top of their chosen branch. If they decide wrongly, their blocks will be rolled back and they lose block subsidies, while having spent the electricity and borne other opportunity costs of not mining well-established, predictable other chain.

In PoS the situation is different: in relative terms, it costs nothing to "forge" (PoS equivalent of term "mine") all possible versions so that block rewards will be retained regardless which chain wins. There are two major kinds of PoS in terms of emission: *inflationary* with money supply constantly increasing through forging, and *re-distributive* where block reward consists only of transaction fees. Therefore, in inflationary kind holder of tokens loses by not participating in forging (being off-line) through the value of his coins being diluted without him harvesting the emission; in re-distributive one the only off-line loss is loss of transaction fees, collected by another forging participant. We strongly prefer re-distributive model among other benefits of Nxt listed in section III, subsection i. To alleviate the problem sometimes referred to as "the early adopters control the system forever", we introduce emission through mining the key blocks on top of re-distributive PoS. The role of key blocks in our system is, however, not limited to emission and will be further discussed in III, iii.

There is "good enough" solution to N@S implemented in Nxt which is to include a reference to Economic Cluster block (ECblock) into each transaction and seal that reference among other important fields with the signature of transaction's issuer [Picard et al, 2013]. Therefore it isn't possible to move large portions of transaction history between competing chains as their ECblocks invalidate; we kept this advantage adapting it to our own needs in the Hybrid, to prevent a rogue miner attempting e.g. a 51% hash-rate attack to fill his unfriendly chain with legitimate transactions performed by the economic majority.

iv. Other PoS Vulnerabilities

The so-called Long Range attack vector [Buterin, 2014-05] is closed in Nxt by limiting MAX_ROLLBACK amount of blocks, which in turn affects the size of permitted re-organization of the chain, and by introducing CHECKPOINT blocks similar to Bitcoin: fixing IDs of the blocks at certain heights in the chain by putting them into the source code (*hardcoding* them). A similar approach, "developer-signed blocks" can be argued against as introducing centralization into the conceptually decentralized system; the problem here is that in a system without PoW component, and even in one with it like our Hybrid but being under 51%-attack, there is no perfectly decentralized way to establish authenticity of the blocks for a new participant who tries to find out which nodes are "truthful" (economic majority-wise) and which are not. The Long Range attack is about obtaining old private keys of the former "whales" (majority holders) who have long since spent their coins and about rebuilding almost the entire chain from scratch, filling it with bogus transactions and thus trying to take control over the system, from the point of view of new unsuspecting participants. Our Hybrid can boast additional protection against this: mining a longer chain, in terms of the Hybrid difficulty is, well, difficult in the sense of PoW.

The Long Range attack and Nothing-at-Stake are two well known, widely discussed flaws of PoS and a heated debate about whether they are real or imaginary threats still rages on. Apart from those, Nxt has it's own: initially weak account IDs and (speculative) hijacking the forging via correctly predicting the online/offline status of active forgers, and abusing the *generation signature* used in Nxt's Blind Shooter PoS implementation [mthcl, 2014]. In our project we renamed GS into *generation sequence*, as it's a 32-bytes hash function value rather than an ECSDA 64-bytes signature, so word "signature" seemed misleading.

Both unique vulnerabilities were closed in our Hybrid: the GS one was closed due to completely unpredictable outside effect of mining: key blocks have their own GS calculated with a variation of Nxt rules; the initially weak account ID is the fact that before the 1st outgoing transaction, if sender of the initial funds who first mentioned the new account on blockchain didn't specify the recipient's public key, the funds being sent are associated only with the eight least significant bytes of the hash of recipient's public key, while the key itself is not published. This allows the brute-forcing of such an account by finding a collision within 64-bit search space, well within reach of current hashing powers and achievable in reasonable time. There is no way to require all Nxt accounts still lacking public keys to acquire them, other than each holder sending an outgoing transaction supplying his/her secret phrase being turned into a private key and generating a signature. The corresponding public key is "announced" and becomes part of the blockchain and the "state" of the system.

In our Hybrid, all account IDs consist of 12 bytes, which makes finding the collisions substantially more difficult. To prevent an attack even further, sender is prompted to provide a public key when account of recipient is new; we inherited this functionality from Nxt and even centralized exchanges known to us (HitBTC) support this to increase security.

III. Project's Motivations

i. The Choice of Nxt

We have chosen Nxt for the following reasons:

- Implemented in Java — the leading industry standard programming language for corporate applications, with a lot of expertise readily available in the community of developers
- Actively used in production and in a state of constant development, the platform has been able to stand the test of time due to its advanced architecture and solid design
- Has been subjected to multiple code reviews by independent experts from various backgrounds, due to it's open-sourced nature
- Allows for fast prototyping and the rapid development of new features
- Present features include, among others: creating and trading assets, setting up conditional transaction execution, sending encrypted messages, registering aliases
- Nxt development team has delivered new features every few months during the last three years, which proves the well-designed and flexible architecture of the platform
- The rich and comprehensive Nxt API which supports over 200 request types has allowed various projects to be built on top of Nxt (e.g. modular UI as a separate layer — for the end user, and test API web interface for the developer)

[Jelurida Swiss SA, 2014]

ii. Pros and Cons of Proof-of-Work

There is a widespread opinion that PoW is the only solution for the "Byzantine Generals' problem", while one of the opposing beliefs is that it's not at all a solution and, in fact, no existing cryptocurrency solves the problem (also known as *power vacuum*) properly. Even if we concur that PoW is good BFT (Byzantine Fault Tolerance), producing blocks in a truly decentralized and scalable manner is computationally expensive. While the network and exchange rate of a PoW token grows, there's a constant arms race between miners, throwing more and more hash-rate at the system to maximize their rewards, raising concerns about electricity use and heat wastage. Once the network growth stops or pauses, "selfish mining" (forming mining cartels to gain unfair advantage, loss of Nash equilibrium starting at 33% of accumulated hash-rate) and even 51% attacks become realistic.

Contrary to the utopian vision of schoolchildren and grannies engaged in issuing Bitcoin on their home computers, the rising popularity of Bitcoin led to the invention of ASICs (specialized chips that can do only hashes, and a lot of them) and mining pools that combine together hash-rate of many individual miners. GPUs previously used in Bitcoin mining were initially reused in the early "altcoins" like Litecoin, but later even so-called ASIC-resistant hashing algorithms acquired their ASICs. At the time of writing, a lot of GPU "rigs" and "farms" stand still since the new coins mined are not worth the electricity.

Increasing the block frequency in order to expand the applications range beyond "digital gold" (Store of Value) slow/expensive system with low throughput, produces orphaned blocks that no-one builds upon, with incentives to out-smart other miners causing their blocks to be orphaned. One solution was to introduce "uncles" (rewarding orphaned blocks with reduced block subsidy) in Ethereum. The rate of orphaned blocks increases once the delay between blocks stops being orders of magnitude larger than the network latency [Buterin, 2014-07]. Another way to allow for frequent transactions is Lightning Network, which places Bitcoin as a settlement layer of a new distributed ledger.

Still another opportunity to implement fast, cheap transactions is a *sidechain*: an additional chain, either not based on PoW or re-using parent's chain work through "merged mining", so that we are saving on computations already done in the parent's chain, with a "trustless peg" — a mechanism to move value to and from the sidechain without giving a chance to a 3rd party to steal it [metro.software, 2018] [Back et al, 2014]. We found this solution particularly useful, and our key block design allows to create pegs to other PoW/hybrid chains with different parameters, features and established markets.

iii. The Benefits of Hybrid Consensus

The opportunity to obtain stake in a hybrid system is technical, not social: all you need is sufficient hash-power, whereas in pure PoS you need current stakeholders willing to sell their stake. Thus at any moment of it's history Hybrid is less of a "closed club" than pure PoS.

Furthermore to have a chance of successful attack

on Hybrid chain, attacker needs to acquire both stake and hash-rate (50% of both for 50% chance of success). Only those with the greatest desire to destroy the Hybrid coin and a few millions of dollars in their account can have a shot at the security of Hybrid blockchain. There is no way to attack only one aspect of security, either PoS or PoW: blocks issued by miners and forgers are intertwined in one Hybrid chain.

In a pure PoW coin, holders do not have a say in the governance and contentious hard-forks like Bitcoin Cash occur as a result of fierce competition between mining cartels; as a future option for our Hybrid, hard-forks can be planned and voted for by stakers and then done automatically if they change only some parameters of the system without requiring new code to be written. Thus the stakers of Hybrid can change or even pause/resume mining by expressing their will. In the current design of Nxt, voting transactions are implemented but a hard-fork would have to be done manually by the developers.

The main reason however to choose Hybrid for our project was to allow for an external "observer" like an Ethereum smart contract to look at compact proofs of PoS history fixated in key blocks, while the key blocks themselves are verified by their PoW "quality". This allows for a smart contract to consume history of the Hybrid and establish true facts about the longest chain, the functionality needed by sidechains.

IV. Results

On top of NxtClone we built a Hybrid PoW/PoS secured Public Blockchain. The active development of the core system, excluding R&D phase and additional infrastructure tasks such as adapting ccminer/sgminer software, took approximately 16 man months costing ~$20K.

The active phase started in March 2018 and in August we had a stable MainNet with hundreds of nodes and blocks each 5 seconds.

V. Discussion

i. Conclusions

Looking back at the history of our project, there was a pivotal moment where we were deciding whether to take BitcoinJ Java language implementation of Bitcoin Protocol and add Nxt-like PoS to it, or to proceed with Nxt-based Hybrid and do the required share-drop in accordance with Jelurida Public License. After choosing the latter we had some problems due to absence of "tips" (Bitcoin stores several alternative histories at once in it's database, whereas Nxt stores only one) and at some point realized that absence of UTXO concept from Bitcoin can make future innovation difficult.

ii. Opportunities for Further Research

In our current design, key block header is modified to contain in addition to a typical Bitcoin-like header: a pointer to previous key block, usually jumping back one or two hundred of "fast blocks" (normal PoS blocks similar as they were in Nxt); a root of "forgers Merkle tree", the structure that enables "observers" mentioned in III, iii to reason about state changes between two key blocks. However, the structure of the tree is allowing only a few applications that we could foresee; in the future, this structure can be made more general. The size of Merkle proofs will become much larger, so such a new tree root may be included in a Coinbase transaction (the 1st transaction of every block, both key and fast, we introduced this for generality from Bitcoin into Nxt-based code) and the old optimized one left in the header. Strict optimizations are required: an observer, implemented as a smart contract, would require a lot of *gas* to operate.

All links are clickable in the PDF version at https://git.io/hybridwithnxt

References

[Nakamoto, 2008] Nakamoto, S. (2008). A Whitepaper. Bitcoin: A Peer-to-Peer Electronic Cash System

[Buterin, 2014-05] Buterin, V. (2014). A blog post. Long-Range Attacks: The Serious Problem With Adaptive Proof of Work

[Buterin, 2014-07] Buterin, V. (2014). A blog post. Toward a 12-second Block Time

[Picard et al, 2013] Picard, J-L. with other contributors (2013). Nxt Whitepaper, built by the Nxt community

[mthcl, 2014] NXT id 5978778981551971141 The math of Nxt forging

[Jelurida Swiss SA, 2014] Jelurida (2017) Jelurida Whitepaper

[Back et al, 2014] Back, A. et al (2014) Enabling Blockchain Innovations with Pegged Sidechains

[metro.software, 2018] Metro Software (2018) Metro Blockchain Whitepaper