Sirius: Reputation-Weighted Proof-of-Stake

M.A. Wiedouw[1], A. G. Lepidis[1]
[1]*Cloudchain, Charlestown, Saint Kitts and Nevis*
getsirius.io – info@getsirius.io

***Abstract*—Current cryptocurrency consensus algorithms raise the maximum number of transactions per second (TPS), but fail to preserve decentralisation by implementing user-elected master nodes. To solve this problem, a novel Proof-of-Stake (PoS) algorithm based on reputation scoring by mining candidate blocks is introduced. Votes are determined by consensus, not by the user. The result is improved resistance to majority attacks and higher transaction speeds. A near-term maximum transaction speed of 4300 TPS is predicted, without decreasing decentralisation more than specified by the CAP theorem, as unreliable nodes are automatically removed from the transaction miner group.**

I. Introduction

The CAP theorem states that consensus algorithms must balance three key factors: safety, liveness and fault tolerance [1]. In a cryptocurrency system, these factors translate to the centralization vs transaction speed vs security trilemma. Current cryptocurrency consensus algorithms suffer from either too much centralization, such as Delegated Proof-of-Stake (DPoS) and Federated Byzantine Agreement (FBA) or transaction speeds that are too low to match that of VISA (2000 TPS), such as Bitcoin [4] [5] [3] [2]. The Sirius protocol solves this by introducing Reputation-weighted Proof-of-Stake (RWPoS), which balances the two factors without introducing more centralization than required by the CAP theorem, while increasing transaction speed. This is accomplished by assigning a reputation score to each node based on its performance in the past. Nodes mine candidate blocks to build their reputation until it is high enough to mine transaction blocks. At the base of this system lies the Proof-of-Stake (PoS) algorithm, a modified version of standard Proof-of-Work (PoW), or Nakamoto consensus.

A. Nakamoto consensus

In Bitcoin, the double-spending problem, which can happen when nodes disagree on the ordering of transactions, is solved with Nakamoto consensus [2]. Here, each node gathers transactions from a peer to peer network into a block. Nodes repeatedly calculate a cryptographic hash of the previous block of transactions and an incrementing nonce, until the resulting hash matches a target, determined by the difficulty. The first node that finds the correct hash includes it in its created block, and propagates it to its neighbours. Forks can happen when there exist competing extensions of the chain. This is resolved when one chain becomes longer before the other one does. The longest chain of blocks, typically containing the highest amount of work, is taken as the correct ordering of events by the network.

B. Proof-of-Stake

Proof-of-Stake, first proposed on the Bitcoin forum, includes a signed transaction in the block hash, thus proving the signer controls the amount of currency in the transaction [15]. The nonce can only be changed every 16 seconds, instead of as often as desired like in PoW. This method is preferable because it prevents a race to ever greater amounts of processing power like in Nakamoto consensus.

C. Smart contracts

Blockchains can also be used for general purpose computation. Smart contracts are pieces of code that are included in blocks and executed, with their output verified by all nodes. Sirius implements the Ethereum Virtual Machine (EVM) platform to enable smart contracts. For more information on smart contracts and the EVM, the Ethereum white paper can be consulted [6].

II. Propagation time

Under ideal network conditions, the number of nodes n in a cryptocurrency consensus group that have received a new block or transaction after r rounds of peer to peer propagation, with the number of nodes to send to at the same time, or outdegree x is:

$$n = (x+1)^r$$

The time t to propagate to n nodes, with a the average time for a random node to propagate a block or transaction to x nodes is:

$$t = ra$$

Combining this yields:

$$n = (x+1)^{\frac{t}{a}} \tag{1}$$

With bandwidth per node b and size s, a is:

$$a = \frac{x \cdot s}{b} \tag{2}$$

Or, setting s and b to 1 ($a = x$) and solving for t:

$$t = \frac{x \log(n)}{\log(x+1)} \tag{3}$$

For positive x, the minimum of this expression is at $x = 1$, which is the optimal outdegree. Note that it is assumed that nodes only propagate to other nodes that haven't received the new block or transaction yet. Because the slow-start property of the regular TCP protocol requires multiple connections be made to reach maximum speed quickly, TCP-BBR which avoids this can be used [7].

A. Model validation

This model can be used to predict propagation times in the Bitcoin network. Approximately 30% of Bitcoin nodes are hosted on cloud providers such as OVH and Amazon [8]. To estimate a, Bitcoin blocks were downloaded and verified with the Bitcoin client using default settings on entry-level instances on OVH and Amazon, and the average bandwidth used were recorded. The bandwidth use doesn't come near maximum capacity because the single-threaded transaction verification algorithm creates a CPU bottleneck [9]. This means that a is proportional to the maximum verification speed of the node. The mean bandwidth used was measured to be (6.80 ± 0.51) Mb/s. This value, together with measurements of block size and the number of active nodes, can be used to predict the time until 50% of nodes have received a new block. The result of the prediction over a 6 month period in 2015, selected because no block size reduction modifications were active yet, is shown in figure 1.

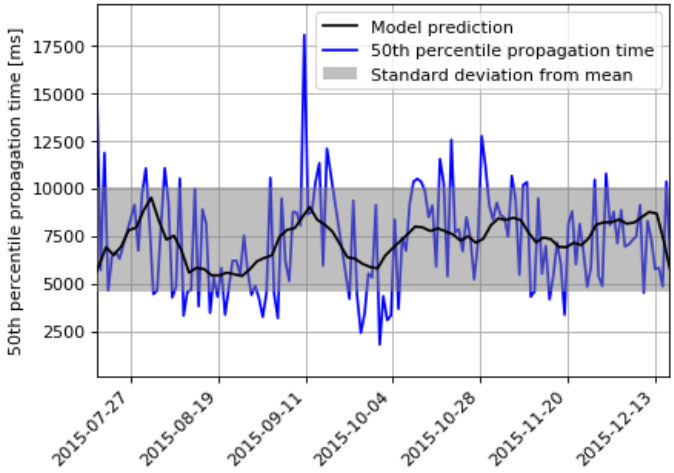

Figure 1. 50th percentile propagation times compared with model predictions. Note that the prediction remains within one standard deviation of the mean of the measurements.

The mean of the prediction over a 6 month period was (7291 $\pm$ 2758) ms, and the mean of the measurements (7382 $\pm$ 1536) ms, for a relative error of 1.25 %. The difference could be caused by measurements of verification speed having been done on higher performance cloud instances than were typically used in 2015. Other possible causes include variability in internet connection quality between nodes and the topology of miner networks.

III. Reputation weighting

Minimizing t in practice can be accomplished by reducing the consensus group size, which reduces t proportional to $\log(n)$ or increasing the verification and transmission speed, which results in a linear reduction. A balanced approach is used, lowering the consensus group size while increasing the verification and transmission speed a. Assuming a homogeneous network, miners can prove that they are honest and reliable and have a sufficiently high verification and transmission speed by building a history of consistently having proposed candidate blocks for a long period. The stake weight value for a group of 32 nodes with the highest reputations is then set to be dependent only on their reputation.

A. Candidate blocks

In RWPoS, candidate blocks are mined in a Proof-of-Stake scheme, but miners are additionally rewarded with a reputation score. The value of this score determines if a node can then participate in the mining of transaction blocks, which continue to contribute to reputation. Candidate block miners serve three purposes: to verify blocks and transactions and relay them, to build reputation to become eligible to enter the transaction block mining group and to serve as a connection point for light clients, who do not verify blocks. For the reputation calculation, the Repucoin method is followed [10].The fraction of mined blocks n to the number of blocks in the entire chain N and consistency defined as the standard deviation of the time delta Δt between proposed blocks σ_t, are used to compute the reputation R of a miner:

$$R = \frac{n}{N(\sigma_t + 1)} \tag{4}$$

Majority attacks, where an attacker attains a large portion of the hashing power in a network in order to then perform a double-spend attack, are usually infeasible in PoS due to the attacker having to buy large amounts of the currency in a short amount of time, which depends on market depth being equal to a significant portion of the total supply. To further constrain such majority attacks, reputation should not immediately begin to grow but should instead be delayed by a sigmoid function to become R_d, where the constant α shifts the function so that reputation starts at zero, and λ is an empirically determined constant that controls the steepness of the curve:

$$R_d = \frac{1}{1 + e^{-(\lambda R - \alpha)}} \tag{5}$$

It can be shown that, because the growth of R is constrained by the fixed blockchain growth ΔN, the growth of R_d only depends on λ [10]. This means that majority attacks can be made infeasible by adjusting this parameter. In this system, nodes that propose many blocks consistently, verify transactions and commit blocks quickly, and produce fewer fork blocks will, over time, gain a higher reputation than others.

B. Forks

The fork rate O, with the block target time T and t the time to propagate to 50% of the nodes, can be approximated as [2]:

$$O = 1 - e^{-\frac{t}{T}} \tag{6}$$

This can be interpreted as follows: for a time t, most of the nodes are working on finding a block which has already been found and is propagating. Their minting is wasted and this lowers resistance to attacks. Because of this, a low ($\leq$ 4%) fork rate is desirable.

C. Algorithm

The RWPoS algorithm can now be described in pseudocode using the definitions in (5) and (4). Four more variables are introduced: The amount of currency a node has staked, or stake weight w, the reputations of all nodes R_i, the honesty factor H, which is 1 by default, but becomes zero when the node has misbehaved by including invalid transactions in a block or relaying invalid transactions, and the boolean value C, which determines if a node can mine the next transaction block if false, and a candidate block if true. Note that the algorithm assumes that each node has a unique reputation score. In practice the miner will identify itself to others by signing a transaction and propagating it on the network.

Algorithm 1 Reputation-weighted Proof-of-Stake

Input: $n, n_r, N, \lambda, \Delta t, w, H, R_i$
Output: w_R, C

$\sigma_t = \sqrt{\frac{\sum_{i=1}^{n_r}(\Delta t_i - \overline{\Delta t})^2}{n_r}}$
$x = \frac{n_r}{N(\sigma_t+1)}$
$R = \frac{1}{1+e^{-(\lambda x-\alpha)}}$
Sort R_i in descending order
if $R \in \{R_0, ..., R_{n-1}\}$ **then**
    C = **false**
    $w_R = HR$
else
    C = **true**
    $w_R = w$
end if
return w_R, C

D. Incentives

All miners must be properly incentivised to ensure optimal network performance. To achieve this, candidate block miners are rewarded with a constant amount c, plus an amount dependent on their reputation. Transaction block miners are rewarded based on their reputation R, scaled by a constant factor c_t only.

IV. Transaction speed

Reputation weighting allows for a substantial increase in transaction speed without decreasing decentralisation more than the minimum caused by a smaller consensus group. Members of this group are also rapidly replaced if their performance degrades for any reason. A consensus group of the 32 highest reputation nodes is selected. The compact blocks proposal (BIP152) is also implemented, reducing effective block size by 98%, by preventing the resending of transactions when they are included in a block [11]. Following advances made by the Bitcoin Unlimited team and implementing multithreaded ECDSA signature validation, a high performance cloud instance can reach 10000 TPS in transaction verification speed [9]. Given that a small transaction is 226 bytes in size, 10000 transactions per second corresponds to a bandwidth b of 226 bytes · 10000 = 2.26 MB/s [14]. Using a 60 second block target time, and a block size s of 59 MB, 59 MB / 226 bytes / 60 = 4351 TPS can be reached. Block size is reduced by 98 %. By equation (3), and setting a to 0.02 · 59 / 2.26, the lowest possible time to reach > 50% of this group is then:

$$t = \frac{1 \cdot \frac{0.02 \cdot 59}{2.26} \log(17)}{\log(2)} \approx 2.13 \text{ s}$$

The fork rate by equation (6) and reduced by BIP152 is:

$$O = 1 - e^{-(2.13/60)} \approx 0.035$$

This means that with a fork rate of 3.5 %, approximately 4300 TPS, more than EOS (4000 TPS) and over double that of VISA (2000 TPS) can be reached while simultaneously making majority attacks infeasible and avoiding the need for user voting mechanisms like in EOS's DPoS and high energy requirements as in Proof-of-Work [4].

V. Conclusion

A novel consensus algorithm using reputation weighting in a Proof-of-Stake setting, as well as a model to estimate the fork rate have been proposed. The proposed fork rate model is validated on data from the Bitcoin blockchain, and then used to predict transaction speeds achievable with the new algorithm while not exceeding a maximum fork rate. The resulting implementation features improved resistance to majority attacks and higher transaction speeds. A near-term maximum speed of 4300 TPS is predicted, with the potential to scale further with better optimized verification implementations.

VI. Future work

The fork rate can optionally be decreased by implementing the greedy heaviest observed subtree (GHOST) algorithm, which also rewards fork blocks [12]. The fork rate S in GHOST is:

$$S = \left(\frac{t}{T}\right)^2$$

This would decrease the fork rate from 3.5 % to:

$$S = \left(\frac{2.13}{60}\right)^2 \approx 0.0126$$

or 1.26 % [13]. A keyblock-microblock system as proposed in Bitcoin-NG, which decouples leader election and transaction confirmation, can further increase transaction speed, with no significant decrease in decentralization and its implementation should be considered.

References

[1] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", ACM SIGACT News, Volume 33 Issue 2 (2002), pg. 51-59.
[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system". http://bitcoin.org/bitcoin.pdf, 2008.
[3] Unknown, "Scalability". https://en.bitcoin.it/wiki/Scalability, accessed April 12, 2019.

[4] Block.one, "EOS.IO Technical White Paper v2". https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, 2018.

[5] D. Mazieres, "The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus". https://www.stellar.org/papers/stellar-consensus-protocol.pdf, accessed April 12, 2019.

[6] Ethereum Foundation, "Ethereum's white paper". https://github.com/ethereum/wiki/wiki/White-Paper, 2014.

[7] Google BBR Team, "BBR Startup Pacing Gain: a Derivation". https://github.com/google/bbr/raw/master/Documentation/startup/gain/analysis/bbr_startup_gain.pdf, 2018.

[8] M. Apostolaki, A. Zohar and L. Vanbever. "Hijacking Bitcoin: Routing Attacks on Cryptocurrencies". IEEE SP. 2017, pp 375-392.

[9] A. Suisani et al. "Measuring maximum sustained transaction throughput on a global network of Bitcoin nodes". https://stanford2017.scalingbitcoin.org/files/Day1/Stanford_2017.pptx.pdf, 2017.

[10] Y. Jiangshan et al. "RepuCoin: Your Reputation is Your Power". https://eprint.iacr.org/2018/239.pdf, 2018.

[11] A. Pinar Ozisik et al. "Graphene: A New Protocol for Block Propagation Using Set Reconciliation". https://people.cs.umass.edu/~gbiss/graphene.pdf, accessed April 12, 2019.

[12] Y. Sompolinsky, A. Zohar, "Secure High-Rate Transaction Processing in Bitcoin". https://eprint.iacr.org/2013/881.pdf, 2013.

[13] V. Buterin, "Toward a 12-second Block Time". https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/, 2014.

[14] J. Belove, "Qtum Mainnet Results December 18 to 24". https://medium.com/@jb395official/qtum-mainnet-results-december-18-24-8e43e51aca3b, 2018.

[15] P. Vasin, "BlackCoin's Proof-of-Stake Protocol v2". https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf, accessed April 12, 2019.

[16] I. Eyal, "Bitcoin-NG: A Scalable Blockchain Protocol". https://arxiv.org/abs/1510.02037, 2015.