# Proof of Stack Consensus for Blockchain Networks

Anjani Barhanpure, Paaras Belandor(✉), and Bhaskarjyoti Das

PES University, Bengaluru 560085, India
pbelandor@nisikitech.com

**Abstract.** The implementations for decentralized consensus within Distributed Ledger Technologies (DLTs) are varied and many, but Blockchain, the underpinning technology underneath Bitcoin has witnessed revolutionary use case success. Although blockchain emerged in the Internet commerce sector as an immutable and decentralized ledger system, it can now be seen as a framework for autonomous decentralized data processing which enforces a flat and open-access network. The primary security threat in Blockchain would be a compromise or fallacy in the Consensus mechanism. The Proof of concept (PoX) approach used in Blockchains has elegantly emulated the leader-election function required in a Byzantine Fault Tolerant (BFT) protocol to simulate the block proposal process. Proof of work (PoW), the consensus mechanism that the Bitcoin protocol employed paved the way for blockchain as a viable DLT for commerce on the internet. Many "alt-coins" subsequently came up, however PoW has saturated with the explosion of popularity of the crypto currency and other alt-coins have achieved sub-optimal solutions. There is the looming "51% attack" for resource pricing algorithms and various other attacks such as Bribe attacks, Sybil attacks and the Nothing-at-stake attack for the other alternate consensus mechanisms. The novel PoStack algorithm is a gamification of the node mining process, which enforces a simple notion: a node's chance of mining crypto currency is proportional to its belief in a node that no one else believes in. The protocol has modest computational and financial needs, which reduces the barrier to entry.

**Keywords:** Blockchain · Proof of Stack · Consensus · Open network protocol · Hybrid consensus · Puzzle design

## 1 Introduction

### 1.1 Advent of Blockchain

The advocation of blockchain has spread into domains outside of Peer-to-Peer tokenised asset transactions. The technology is however proclaimed for this use case through the success of Bitcoin and other crypto currencies. Blockchain technology is an assimilation and reuse of many granular elements of cryptographic


S. M. Thampi et al. (Eds.): SSCC 2018, CCIS 969, pp. 104–116, 2019.
https://doi.org/10.1007/978-981-13-5826-5_8

primitives. The essential innovation in Blockchain is the way in which the technology modelled the replicated state machine principles for distributed nodes in a way that permitted a scalable network with its features of information propagation, openness and tamper resilience. Moreover, the notion of decentralization and distributed consensus has heralded blockchain as a decentralized Virtual Machine for token driven Applications. With the advent of Web 3.0 in the recent years, Decentralized Applications or "DApps" based on Distributed Ledger Technologies are at the forefront of this revolution. Studies and the upsurge of white papers of blockchain based applications have established blockchain as one of the most disruptive realignment of digital ecosystems from FinTech to subscription based services and IoT.

### 1.2 Amalgamation of Distributed Consensus Principles

In the domain of distributed consensus, the PAXOS [5] algorithm first described in 1998 is one of the longest running explicit leader-based implementation of distributed consensus. Its complexity surpasses its usage and so subsequent research and implementations of distributed consensus algorithms can be seen essentially as variants of PAXOS or derivatives of it. A notable variant of PAXOS, known as RAFT [8], developed in 2014 has salient features of being an understandable algorithm, with a refined leader election process and fault tolerance capabilities. Leaders in such systems send out the blocks with confirmed transactions. The problem here is the possibility of a DDoS attack on the leader. "Follow the leader" is a well known issue with leader based system, bot nets can keep DDoS-ing the current leader and crashing the network. Although these algorithms portrayed the properties needed for an ideal decentralized system, these never carried enough capabilities to enable them to be employed in an open network.

Even older still, are Voting based consensus algorithms that have sound mathematical backing and literature that stretches back decades. The idea that percolated into actual applications that were deployed were hybrid versions of this. Pure voting based systems are almost never deployed as the overhead in broadcasting votes is an exponential build up of messages and any circumvention of this shortcoming still fails to project this type of algorithms in its purest form from becoming a viable solution for distributed consensus. The research in this topic has inadvertently spurred more conscious research in more efficient directions.

With the amalgamation of research in this space of distributed consensus, the issue of distributed consensus in an open network architecture gained more attention. The Byzantine General's Problem [6] first stated around 1982, was used to explain the problem of malfunctioning/malicious nodes that give conflicting information to different parts of the network. This modelling of dishonest nodes in a network gathered momentum for researchers to architect systems that were resistant to these conditions. Byzantine Fault Tolerance is the key idea behind creating a scalable open network of trust-less nodes that can be synchronized and tamper proofed. In 2008, an anonymous person or group came out with

a solution for electronic cash called Bitcoin [7]. This was the first consensus mechanism that had economic deterrence in the form of Proof of Work. Security problems ensued over time: Malicious forking of the blockchain; Overpowering of computational resources using Application-Specific Integrated Circuits (ASICs); Bribe attacks; Firewall attacks; Fairness issues. The resilience of Bitcoin and other blockchains powered by PoW is due to the various adjustments made to the protocol over time.

Economy based consensus mechanism began to come up after the saturation of PoW became more evident. Here, the consensus algorithm itself tries to simulate the way an economy works. Factoring in the chaotic system that has stochastic eventualities and instabilities, economy based blockchains possess the uncertainty factor as well as the mass mentality factor: if everyone has a financial incentive to add on to the longest block, and if most nodes emulates this success oriented behaviour, every other node follows. However, these systems are hard to build as secure markets. The idea of an economy based blockchain that could deter any attack such as the "Nothing at stake" problem has built up more research into this form of consensus protocols that is referred to as Proof of Concepts (PoX) [11], where the Concept aims to build an economy that can inherently deter attacks through gamification of the mining process.

This paper covers the summary of relevant literature that leads to the formulation of the novel Proof of Stack algorithm in Sect. 2. Following this, Sect. 3 describes the approach of the Proof of Stack algorithm and the mathematical operations involved in its operation. Section 4 reports the Results and Best case evaluation of the algorithm, focusing on fairness for all nodes and being fault tolerant. Section 5 describes a potential architecture designed around Proof of Stack, with an added architectural design to enable scalability of the blockchain network, bringing in the concept of constituencies and block bubbling. Section 6 covers the impact of the algorithm and its ramifications in the current landscape of Distributed Ledger Technology.

## 2 Related Work

The benchmark paper by Satoshi Nakamoto in 2008 highlights the "Double Spending" problem and came up with the first viable solution for an electronic cash system. The ideology and pattern of design was derived from previous work, specifically Hash-cash and b-money. The "Proof Of Work" distributed system enforced the security of the network and is still used to this day as the go to consensus mechanism for blockchain based "dApps". The paper highlighted the structure and design of the blockchain and highlighted the boundary condition for honest nodes to adhere to the network.

The idea of security in a decentralised system is modelled from the Byzantine General's Problem and in the purview of blockchain networks these characteristic failures can cause faulty nodes to exhibit arbitrary or deviant behaviour such as malicious attacks/collusions and double spending attacks; node mistakes and network connectivity issues. With the bifurcation of blockchain networks into

Permissioned and Permissionless, the latter has a more slacking adherence to the conventional Byzantine Fault Tolerance properties and allows for no node identity registry or explicit hierarchy and synchronisation. Therefore, consensus protocols in this realm need to be scalable, while still being tolerant to pseudo identities.

### 2.1 Advent of Proof of Concept Mechanisms

In order to tackle security, scalability and synchronous issues in permissionless blockchain networks, the Proof of Concept (PoX) scheme became the de facto design choice. A comprehensive survey into the disambiguation and classification of the various Proof of Concept protocols have been studied which also brings into light the concepts of non interactive Zero Knowledge Proofs (ZKF) which is the basis for Proof of Concept protocols.

In Bitcoin, the "Solution-Verification" class of Proof of Work protocols [2] was employed. Here, the self-imposed challenge and solution, parameterised by network activity, is provided by the block proposer and on verification of the solution, the block is accepted until there is emergent consensus from a majority of nodes. This scheme assumes an altruistic behaviour in terms of information forwarding. This results in two points of view: From the node perspective, every node is engaged in a cryptographic block proposal race and from the perspective of the network, an implicit function of leader election and information propagation is enforced. The PoW scheme is computationally expensive and is prone to the "51% attack" from the upsurge of oligarchy in terms of processing power. The current landscape of its usage suggests that the protocol has saturated it's overarching security design.

Privy to the shortcomings of PoW, many alt-coins were designed with their own PoX schemes. The idea behind decentralised consensus in permissionless blockchain networks remained the same: nodes in the blockchain network had to non-interactively prove their stake or commitment to specific quantifiable resources (in the case of PoW, it was hash-rate); moreover the network as a whole should have a stochastic function to yield the subsequent leaders for block proposal. Proof of Stake [4] was proposed as the underlying consensus mechanism in "Peercoin". The notion of "virtual mining" was brought into light; the design proposed that security against the malicious or arbitrary modification of transaction ordering is maintained by locking a node's proposed coin amount as a stake or vested interest in ensuring the system maintains its integrity. This ensures that all nodes conduct themselves in a non malicious manner as they have the most to lose if the network is polluted with malicious or arbitrary transaction ordering. Peercoin also factored in the metric of coin age, which weights the miner's (rather minter's) chance for the puzzle solution. The initial distribution of coins to mining nodes has had several alternate approaches including the Ouroboros protocol [3] and PoW infused hybrid variant known as Proof of Activity. With the careful design and circumvention of security threats, the "Nothing at Stake" attack and the "Grinding attack" persist that threaten the network. Proof of Stake provides insight into resource backed mining in the form

of virtual mining. The strength of the protocol is weighed down by the lack of fork tolerance in the protocol, which reveals itself to be the primary breach point for attacks.

Proof of Concepts associated with commitment or adherence to specific resources such as Proof of Space and Proof of Elapsed-Time require the possession of hardware resources dedicated to the mining process as a form of vested interest.

Proof of Burn [9] involves a partial virtual mining scheme whereby nodes send their coins to an address which is un-spendable, essentially burning the coins for ever. This idea emulates the activity of a mining rig used by nodes in PoW which churns electricity, hardware and time by pinning this cost to the act of burning coins. This system is incorporated into "Slimcoin". The pertinent issue with Proof of Burn, similar to Proof of Stake is the notion that the rich get richer. Proof of Burn does distribute currency in a fair and decentralized manner, however it's primary use case can be seen in seeding off new currencies.

### 2.2 Overview of Security Threats and Comparison Criteria

Before drilling down on the Comparison of consensus algorithms, it is necessary to understand the different unique attack vectors possible on a blockchain so that the security of the algorithms can be appreciated. Some of these attack vectors identified by a study [1] by the Bitfury Group are presented below:

**Denial of Service (DoS).** Wherein, you send nodes high volume of transactions that prevent them from working on the legitimate ones. Distributed DoS is a variant of this form of attack.

**51% Attack.** If an attacker controls more than 50% of the nodes, then he can influence the consensus process. He can alter the blockchain by creating a fork, by enforcing the acceptance of a block with manipulated transactions.

**Double-Spend.** In cryptocurrencies, this is a case when the same coin is used for multiple transactions. Sybil Attack. When a node assumes multiple identities and tries to pass itself off as multiple nodes in the network.

**Cryptographic Attack.** The oligarchy and honing of super computing capabilities in the hands of a few. This shifts power into the hands of a few.

**Byzantine Attack.** A single or few nodes prevent consensus.

Finally, a report by KPMG [10], highlights the present status of consensus mechanisms and their different use cases. The report provides a detailed overview of the criteria to compare algorithms with and a questionnaire to decipher the features required from a blockchain based on a business use case. Using the parameters mentioned in this report, a comparison of consensus mechanisms is provided.

There are a wide variety of algorithms for attaining consensus and they are defined or identified by the following parameters as shown in Table 1.

These parameters can be used to qualitatively classify consensus protocols. For instance, Proof of Work and Proof of Burn are associated with an

**Table 1.** Parameters for qualitatively classifying consensus mechanisms

| Parameter |
|---|
| Decentralized Governance |
| Quorum Structure |
| Authentication |
| Integrity |
| Non-Repudiation |
| Privacy |
| Fault Tolerance |

"Open or Permissionless" governance, whereas DPOS has a "Partially centralized, Permissioned-Governance". However, the core criteria for the analysis of consensus mechanisms is better approached by looking at the Performance metrics:

1. **Throughput.** Volume of transactions the DLT is able to process
2. **Latency.** How long the DLT takes to confirm and commit each transaction
3. **Scalability.** How many nodes can the DLT support without compromising performance
4. **Security.** How resilient is the DLT system
5. **Cost.** How much it costs to build/run/join the system.

## 3 Approach

The main concept, involves a betting round, where each node places a bet (as a Stack of their coins) as a sense of belief in another node. Modelled as a Nash equilibrium where each node aims to bet on other nodes that no one else bets on; it enforces a collusion resistant system while also establishing fairness among nodes in mining likelihood by weighing in the element of pure chance.

Proof of Stack can be analogously thought of in two ways: it is a variant hybrid of Proof of Stake (Virtual Mining) and Proof of Burn (Locking crypto to another address); it is similar to the operation of TF-IDF (Term frequency-Inverse Document Frequency) in Natural Language Processing, specifically in n-grams.

### 3.1 Central Notion of the Algorithm

This algorithm is designed such that "a node's chance of mining crypto currency is proportional to its belief in a node that no one else believes in". The reason for this kind of incentive is to create fairness among reward distributions, whilst maintaining participation in mining from all nodes. As was interpreted from predecessor algorithms, fairness in mining was not reproducible over the long run. With respect to PoW, the mining power became an oligarchy. In PoS,

stakeholders with larger stakes become more likely to win in further stages, a symptom of "The rich getting richer". Fairness is achieved in Proof of Stack by introducing the notion of Stack-Worth intuition, a gamification of the mining process. The implicit leader election function chooses the winner of the current betting round as the node that propagates the next block to all other nodes, who then validates its ordering. Fairness is achieved by not restricting reward distributions to nodes with high stake or mining power.

The algorithm is a puzzle design, based off the work by Wenbo Wang, Dinh Thai Hoang, Zehui Xiong, Dusit Niyato, Ping Wang, Peizhao Hu, Yonggang Wen. The work specifies that a complexity gap needs to exist in any puzzle design which is easy to verify and hard to invert/re-solve. For open blockchains, there needs to be a non-interactive ZKP (Zero Knowledge Proof) as the verifiable random function. Following this work, Proof of Stack is modelled as a PoX process as shown in Table 2.

**Table 2.** Three stage abstraction of a PoX scheme

| Stage | Description |
|---|---|
| Initiation (Generator of random seed or keys) | Open to Design Choice. Needs to provide a common reference string. One option could be the Oracle querying scheme, where the oracle behaves like a beacon, seeding off random values for calculation purposes |
| Execution (Challenge and Proof Generator) | Happens in two rounds. First round initiates the betting sequence. The second round obtains a winner based on the algorithm. Either the previous leader takes up the responsibility of handling these calculation or it can be outsourced to the Oracle again |
| Verification | Verification is of two types: Verifying that the sender actual has enough stake and Verifying the calculations of the algorithm for the current cycle |

### 3.2 Working of Proof of Stack

**The algorithm consists of two parts:**

– **Betting Stage.** In this stage, each node bets on every node that is currently active. The reason that the elements being bet on are other nodes is because of scalability. As more nodes join the network, there has to be a wider distribution of probability between participant miners. An arbitrary upper limit is

needed to scale accordingly and taking the number of nodes as the number of possible bets creates an implicit function for scaling. The bet placing process is subject to a lot of empirical testing and theoretic game analysis. The latter has been illustrated in Sect. 4.

– **Results Stage.** In this stage, all the calculations needed to declare the winner is done. The results stage can be executed by the previous leader or by an oracle (data triggers or data feeds). The winner is chosen by Roulette Wheel selection of their Stack-worth probabilities, which emphasizes the notion of an intuition that is unique to each node. This intuition plays into the protocol as a sense of randomness that is accessory to the initial sense of weighted stake.

**Example Illustration.** Let us assume we have three nodes: A, B and C. Each node places a bet on the other nodes as follows.

**Table 3.** The PoStack matrix of bets, PoStack

| | A | B | C |
|---|---|---|---|
| A | 0 | 2 | 3 |
| B | 2 | 0 | 3 |
| C | 2 | 3 | 0 |

No node can bet on itself. The row vector would sum up to how much a node bets in total. The column vector would sum up to the total sum being bet on the particular node.

For each value in the matrix (i, j), the following calculation is made/metrics are used:

$$Stack = The\ amount\ node\ `i'\ bets\ on\ node\ `j',\ PoStack[i,j]$$
$$Worth = How\ rare\ is\ it\ for\ a\ node\ to\ bet\ on\ j$$

Hence *Stack is the values within the matrix itself. Refer Table 3.*

Worth is calculated as the total bet amount placed by all nodes divided by the sum of bets placed only on node j. Stack is a function of X and Y, while Worth is a function of only Y.

$$Stack\text{-}Worth = Stack * Worth$$

Stack-Worth is high when Stack (bet amount on a node) is high and Worth (how few other nodes also bet on the same node) is high.

Stack Worth gives us values that can be used as a weight to calculate the probabilities to decide which nodes are rewarded.

The calculation of Stack-Worth is given in Tables 4 and 5. Since we have to use these values in the matrix to give us weighted probabilities, we normalize them. This results in a Normalized Stack-Worth Matrix as shown in Table 6.

**Table 4.** Calculation of Stack-Worth

| Betting node X | Betting on Y | Stack (X,Y) | Worth (Y) | Stack-Worth |
|---|---|---|---|---|
| A | B | 2 | 15/5 | 6 |
| A | C | 3 | 15/6 | 7.5 |
| B | A | 2 | 15/4 | 7.5 |
| B | C | 3 | 15/6 | 7.5 |
| C | A | 2 | 15/4 | 7.5 |
| C | B | 3 | 15/5 | 9 |

**Table 5.** Stack-Worth Matrix

| | A | B | C |
|---|---|---|---|
| A | 0.0 | 6.0 | 7.5 |
| B | 7.5 | 0.0 | 7.5 |
| C | 7.5 | 9.0 | 0 |

**Table 6.** Normalized Stack-Worth matrix

| | A | B | C |
|---|---|---|---|
| A | 0.0 | 0.13 | 0.17 |
| B | 0.17 | 0.0 | 0.17 |
| C | 0.17 | 0.20 | 0.0 |

We perform Roulette Wheel Selection using these normalized values as the weighted probabilities. The "better" and the "bettee" (node being bet on) selected randomly are then rewarded for their mining efforts.

## 4 Results and Evaluation

While designing this algorithm, there are several design issues that were considered. The most important one was: Should the amount that any node can bet totally across the nodes in the system be fixed or variable? From the results in this sections we can see that giving each node a fixed amount to distribute as bets is better than letting them bet any total amount of their choice.

Looking at this from another angle we can classify the results as follows:

1. Equitable Spending (Betting on all nodes equally):
   - Favours no one if fixed amount;
   - Favours the richer if variable amount.
2. Non-Equitable Spending (Betting on all nodes unequally)
   - If everyone goes full risk mode:
     - Favours no one if fixed amount

**Table 7.** Case-wise evaluation of node behaviour

| Case | Archetype of node | Fixed amount of spending | No constraints on spending |
|---|---|---|---|
| **Case A:** All nodes bet equal stack on every other node | Any Node | Equiprobable | Favours the richer |
| **Case B:** Each node places its full stack on only one other node | Nodes with Heavy Stake | Equiprobable (Richer node burns more stack for no reason) | Equiprobable (Richer node burns more stack for no reason) |
| | Nodes with Light/No Stake | Equiprobable (Favourable to poorer as their chances remain the same for lesser stack burn) | Equiprobable (Favourable to poorer as their chances remain the same for lesser stack burn) |
| **Case C:** Mixed Behaviour | Nodes with Heavy Stake | Favours based on Stack-Worth intuition | Favours based on Stack-Worth intuition |
| | Nodes with Light/No Stake | Favours based on Stack-Worth intuition | Favours based on Stack-Worth intuition |

    - Favours no one if variable amount (Burns out the richer)
- If mixed mode:
    - Favours the node who was majority investor in a node that was least bet on.

Specifically, from point 1, we can see that if this was the case (variable amount), then the system would end up mimicking the Proof of Stake Algorithm. Since this was sub optimal in the sense that monopoly over mining is created for the richer nodes (those having more stake in the system) it does not provide a fair chance of competition for new incoming nodes. Hence, the design decision taken was to give each participating node a fixed amount to bet (Nodes can only participate in mining if they have the required balance). Refer Table 7 for a case-wise evaluation of node behaviour.

## 5 Hybrid Leader Based Proof of Stack

The paper proposes a hybrid algorithm, combining the Proof of Stack algorithm with a hierarchical leader based architecture to allow for scalability. This design does bear the caveat that it introduces varying degree of centralization. The architecture's salient features are described in the (see Fig. 1). This section highlights the integration of the Proof of Stack algorithm into a specific blockchain infrastructure.

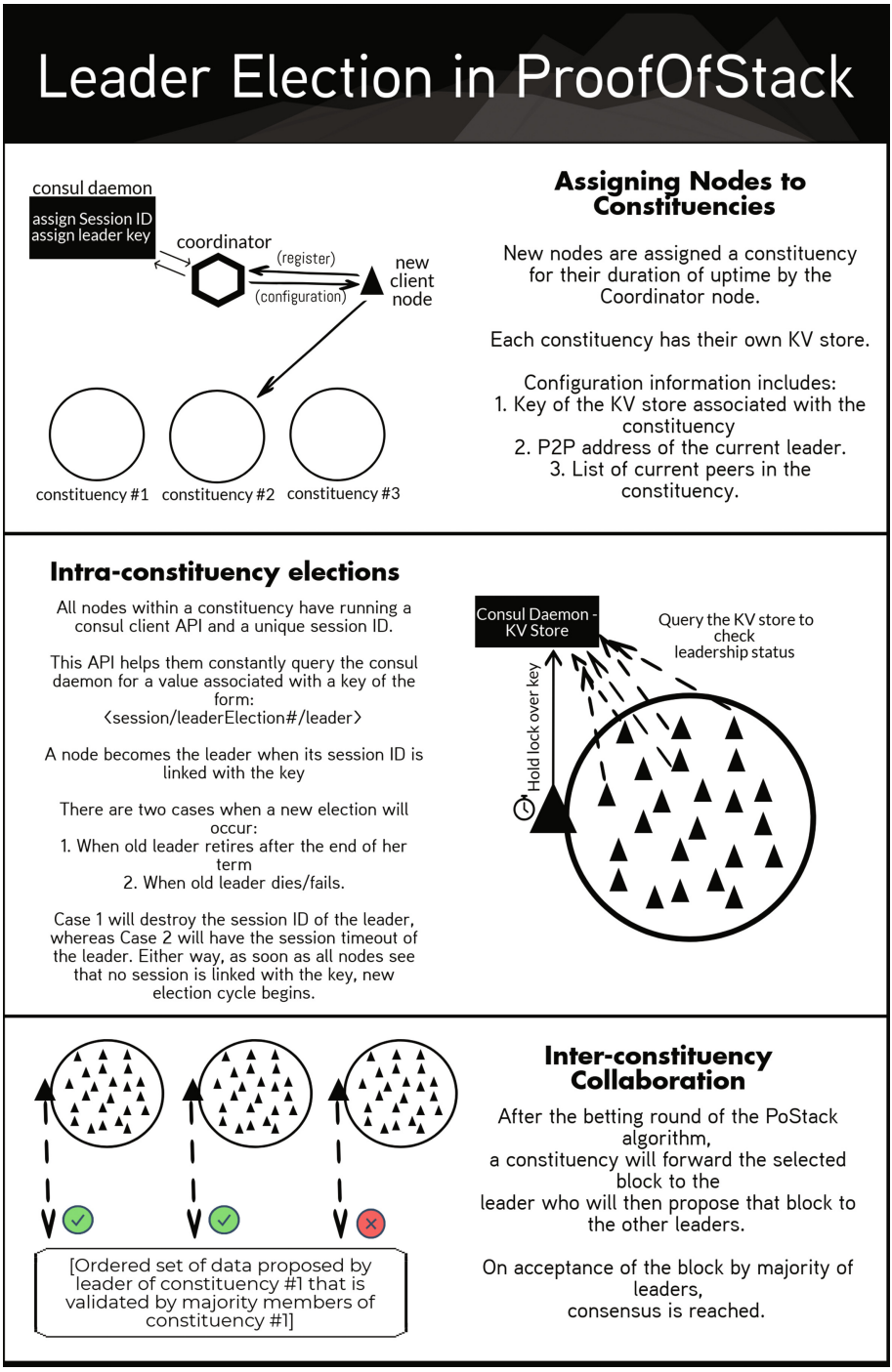


**Fig. 1.** Features of the Leader-based architecture that incorporates the notion of "constituencies" and bubbling block proposals to arrive at consensus with a scalable architecture.

**Table 8.** Issues in conventional leader based consensus

| Issues in conventional leader based consensus | Solution |
|---|---|
| **DDoS Attacks:** A compromised or malicious node may overload the leader with transactions, forcing her to shut down. Then the node "follows the leader" and continues to do so for all succeeding leaders | The proposed architecture has no exclusive copy of the ledger with the leader and the PoStack winner selection is easy to verify, therefore DDoS-ing the leader will not hinder the once started betting round as nodes are self- reliant. Moreover, if one constituency goes down, consensus can still be reached |
| **Unfairness in leader's responsibilities:** The leader gets to choose the order of transactions, when to broadcast and commit etc. i.e. too much power to the leader | The leader has no such responsibility here. Merely a nominal head and a broadcasting point |

As opposed to the general use cases of Leader Based consensus such as in PAXOS or RAFT, here the leader is not the one responsible for coordinating the actual values of the distributed ledger. Here, the leader has essentially the following duties:

1. Quicken the broadcast of transactions to the constituent members
2. Coordinate and select winner of the betting round
3. Broadcast block to other leaders for final signing off.

Some of the cons of leader-based mechanisms and how the mechanism proposed overcomes it is shown in Table 8.

## 6 Conclusion

The research has established an innovative approach to reaching agreement between nodes in a decentralized system. It does so by combining two techniques i.e. Leader based hierarchical approach of network architecture and Novel Consensus Algorithm: Proof Of Stack.

The Leader based hierarchical approach of the network architecture allows us to scale the architecture and make it fault tolerant. The Proof of Stack algorithm eliminates the need for wasteful computations and prevents stakeholders from taking over all rewards of the network by creating a random probability based on betting weights. Hence, every node tries to get the most bets while simultaneously trying to place a bet on the node, which might have gotten the least bets. All attempts to increase a node's chances of winning the round play out similar to a zero sum game among the nodes in the network. Every node wants to make the others believe they should bet on them and hence every node is participating in this game.

The consensus mechanism proposed requires extensive testing with an appropriate quantity of hardware devices across a global network running the PoStack protocol to participate in test trials and generate metrics to quantitatively assess the efficiency of the proposed algorithm. Further action steps also include redesigning the current software simulation of the consensus mechanism and finessing the Proof of concept application built atop the protocol stack.

## References

1. Bitfury: Proof of Stake versus Proof of Work (2015), (White paper). https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf
2. Jakobsson, M., Juels, A.: Proofs of work and bread pudding protocols (extended abstract). In: Preneel, B. (ed.) Secure Information Networks. ITIFIP, vol. 23, pp. 258–272. Springer, Boston, MA (1999). https://doi.org/10.1007/978-0-387-35568-9_18
3. Kiayias, A., Konstantinou, I., Russell, A., David, B., Oliynykov, R.: A provably secure proof-of-stake blockchain protocol. IACR Cryptology ePrint Archive 2016, 889 (2016)
4. King, S., Nadal, S.: Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, 19 August 2012
5. Lamport, L.: The part-time parliament. ACM Trans. Comput. Syst. (TOCS) **16**(2), 133–169 (1998)
6. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. ACM Trans. Program. Lang. Syst. (TOPLAS) **4**(3), 382–401 (1982)
7. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
8. Ongaro, D., Ousterhout, J.K.: In search of an understandable consensus algorithm. In: USENIX Annual Technical Conference, pp. 305–319 (2014)
9. p4titan: Slimcoin.org A Peer-to-Peer Crypto-Currency with Proof-of-Burn "Mining without Powerful Hardware" (2014), (White paper). https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/whitepaperSLM.pdf
10. Seibold, S., Samman, G.: Consensus: immutable agreement for the internet of value. KPMG (2016). https://assets.kpmg.com/content/dam/kpmg/pdf/2016/06/kpmgblockchain-consensus-mechanism.pdf
11. Wang, W., et al.: A survey on consensus mechanisms and mining management in blockchain networks. arXiv preprint arXiv:1805.02707 (2018)