# A COMPARATIVE FRAMEWORK FOR EVALUATING CONSENSUS ALGORITHMS FOR BLOCKCHAINS

Dipti Mahamuni

Ira A. Fulton Schools of Engineering – School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

***ABSTRACT***

*The past five years have seen a significant increase in the popularity of Decentralized Ledgers, commonly referred to as Blockchains. Many new protocols have been launched to cater to various applications serving individual consumers and enterprises. While research is conducted on individual consensus mechanisms and comparison against popular protocols, decision-making and selection between the protocols is still amorphous. This paper proposes a comprehensive comparative framework to evaluate various consensus algorithms. We hope that such a framework will help evaluate current as well as future consensus algorithms objectively for a given use case.*

***KEYWORDS***

*Consensus Algorithms, Blockchain, Comparative Framework, Decentralized Ledgers.*

## 1. INTRODUCTION

The last five years have seen an unprecedented increase in the number of projects for decentralized ledgers. Primarily, what differentiates one project from another is its core consensus algorithm. What started with the first generation of proof of work systems such as Bitcoin and Ethereum have given way to newer generations of systems that include proof of stake, proof of elapsed time, proof of authority, and other DAG-based consensus systems.

Each of these algorithms features some unique differentiators that are analyzed in literature. Some of these analyses only focus on comparing a handful of algorithms [1], while other studies present the comparative analysis based on a single feature such as performance, security, or cost. For example, Cao et al [2] compare various categories of proof of work, proof of stake, and DAG-based algorithms based solely on their performance characteristics.

This paper presents a comprehensive framework for comparing various consensus algorithms to objectively evaluate various consensus algorithms based on the needs of the project.

The rest of the paper is organized as follows. In section 2, we present the evaluation criteria for comparing consensus algorithms. Section 3 illustrates the decision-making framework and its limitations. Future applications extending this work are presented in Section 4. The paper concludes in the subsequent section.

David C. Wyld: CSITY, NWCOM, SIGPRO, ASOFT, AIFZ, BDIoT, ITCCMA, CLSB, DTMN, MLNLP - 2021
pp. 115-123, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111411

# 2. EVALUATION CRITERIA

In this section, we present each evaluation criteria for the proposed framework and discuss its importance.

## 2.1. Security

The basic premise of a decentralized ledger is predicated on the security of the consensus algorithm. If an attacker could compromise the security, trust in the entire blockchain can be diminished.

### 2.1.1. Sybil and Eclipse Attacks

Douceur described the Sybil attack in 2002 [3] where a malicious entity could gain a large influence on the blockchain network by creating a large number of fake identities, devices, IP addresses, or virtual machines. While the Sybil attack tries to subvert the network as a whole, the Eclipse attack tries to prevent an honest node from obtaining the current information by surrounding it with malicious peers.

The current set of consensus algorithms (proof of work, proof of stake, etc) are resistant to Sybil and Eclipse attacks as long as a malicious actor does not get control over a large proportion of hashing power, stake, or the number of DAG nodes.

If this cannot be ensured, then these protocols can fall victim to the Sybil attack [4]. Once the attacker fills the network with malicious clients that are under his control, he possesses control. When all clients work in accordance with the attacker, proof of work is compromised.

A good consensus algorithm must be resistant to these attacks.

### 2.1.2. 51% attacks

If an attacker can control the majority of the hashing power in a network, then they can control the production of blocks in the network. They can create fraudulent entries in the ledger or prevent legitimate transactions from being recorded in the ledger.

The proof of work algorithm as well as multiple variations of proof of stake algorithms (e.g., basic proof of stake, delegated proof of stake, leased proof of stake, etc.) are particularly vulnerable to the 51% attack.

Sayeed and Marco-Gisbert [5] conclude that in most cases, security techniques usually cannot protect against the 51% attack because the weaknesses are inherited from the consensus protocol. However, a good algorithm must prevent attackers from rewriting history on the blockchain. In the least, the algorithm should make it trivially transparent as to which of the 51+ % of consensus nodes are in agreement while committing fraud.

### 2.1.3. Internet-based attacks (aka routing attacks, BGP hijacking attacks), DDoS attacks

The attacker uses the internet protocol (IP) and the associated routing protocols such as the Border Gateway Protocol (BGP) to divert traffic away from honest nodes. The attacker can then prevent publication of the generation of honest blocks and push malicious blocks or partition the network to create a double spending attack.

The lack of a trusted messaging system is core to the classic Byzantine General Problem. Even the earliest proof of work algorithm in the Bitcoin network was designed to be a probabilistic solution to this problem as written by Satoshi Nakamoto [6]. However, there have been several successful attacks on these networks over the years.

In addition, a distributed denial of service attack can either take down honest nodes or create artificial partitions in the network.

A good consensus algorithm must show strong resistance to Internet-based attacks. More importantly, the behavior of the network is important when such an attack is underway. While it is acceptable for the network to stop processing transactions when under attack, a good consensus protocol should ensure that an attacker can never compromise the integrity of the ledger. Leaderless DAG-based algorithms perform better than the leader-based algorithms in this regard.

#### 2.1.4. Double Spend attacks

Double spend attacks allow a holder of an asset to spend it more than once. This can happen in multiple situations such as an attacker using a race condition between two transactions before they are finalized on the blockchain (race attack), the merchant not verifying a sufficient number of blocks for finality (Finney attack), or the attacker creating a fraudulent block that is not confirmed on the blockchain.

Both proof of work and proof of stake algorithms present vulnerabilities for double spend. In a slow proof of work system, an attacker can use the time required to create multiple blocks to launch a double spend attack. The proof of stake algorithms are vulnerable to double spending attacks due to a problem called “nothing at stake” [7]. This means that if a malicious node has nothing in its stake, then it has nothing to lose and nothing to counteract its malicious actions.
A good consensus algorithm should demonstrably prevent double spend attacks.

### 2.2. Decentralization

Decentralization is critical to the operation of any blockchain. Without decentralization, the blockchain degenerates to a centralized ledger or a database and is therefore susceptible to manipulation by a single or a small number of organizations.

The selection of an algorithm directly influences the degree of decentralization. For example, proof of stake’s miner selection is done based on the assets, or amount of cryptocurrency, that a miner owns. Because of this, the algorithm is prone to becoming centralized over time, allowing richer accounts (known as whales) to have more control over the blockchain [7].

The delegated algorithms (such as Delegated Proof of Stake or Delegated Byzantine Fault Tolerant algorithms) typically elect delegates who must reach consensus to verify and add a block to the blockchain [7]. Since there are a limited number of delegates, there is a risk of the system becoming centralized.

Let us look at other factors that affect the decentralization of a consensus-based network.

#### 2.2.1. Decentralization through scale

The scalability of the nodes that run the ledger is important to the network. The trustworthiness of a network increases substantially as the number of nodes that run the network increases. However, this requires extra communication and additional time to reach consensus.

A good consensus algorithm must be able to scale to a large number of nodes. Typically, there are tradeoffs between the extra security and trust added by scaling to a large number of nodes and the communication and latency overheads brought on by large-scale networks.

#### 2.2.2. Geographical distribution of nodes

Though this is not necessarily a consensus protocol design issue, if any aspect of the consensus protocol encourages the concentration of nodes in one country or a geographic area, then the trustworthiness of that ledger is reduced.

For example, a disproportionate number of Bitcoin blocks are mined out of a few countries due to the relatively cheap electric power required for power-hungry proof of work algorithms. A good protocol should prevent such geographical concentration from occurring.

#### 2.2.3. Permissioned versus Permissionless

A good consensus protocol should be able to run in a permissionless environment. It should be possible for anybody in the world to validate the transactions on a blockchain without explicitly asking permission from the existing set of nodes.

#### 2.2.4. Open source - decentralization of development and source code

The consensus algorithm as well as the source code should be open source. Any security vulnerabilities can be easily found by the open source community. Similarly, this eliminates over-dependence on a few smart engineers for continued enhancement of the algorithm.

#### 2.2.5. Requirement for a specialized hardware

Some consensus algorithms require specialized hardware to function effectively. For example, consensus algorithms based on Proof of Elapsed Time [7] rely on specialized hardware present on Intel CPUs (SGX) supporting Trusted Execution Environment (TEE). Similarly, a lot of Ethereum mining is accelerated using specialized GPUs today.

A good consensus algorithm should ensure that it does not need any specialized hardware. More importantly, it should ensure that the presence of specialized hardware, including but not limited to GPUs, FPGAs, and ASICs, should not provide any unfair advantage in creating blocks on the blockchain.

### 2.3. Scalability

#### 2.3.1. Energy Consumption

The proof of work algorithm requires a lot of computational power for a miner to be able to add a block to the blockchain. This computation uses an excessive amount of electricity as compared to proof of stake [4]. The proof of stake algorithm reduces computational power, hence reducing

energy consumption. Panda et al [4] also conclude that the proof of burn algorithm has a better energy consumption rate than proof of work does.

A good consensus algorithm should process transactions with minimum electricity consumption.

#### 2.3.2. Finality

The time to finality of a transaction is defined as the elapsed time from when a transaction is submitted to the network to the time the transaction is recorded in the blockchain. Traditional proof of work algorithms lack a strict definition of finality since waiting for more blocks in the blockchain only increases the probability of the transaction being final. Chaudhry & Yousef [8] present a table that lists probabilistic versus deterministic finality times of various algorithms. Since the proof of stake algorithm spends less time doing complex computations, block finality time is faster than it is for proof of work [4]. Newer DAG-based consensus algorithms have further reduced this time to a few seconds.

Many practical applications, such as credit card transaction processing, require fast finality times. A good consensus algorithm should support finality in a few seconds.

#### 2.3.3. Throughput (Transactions per Second)

Large networks that have high transactions per second cannot use the proof of work algorithm due to its time-consuming nature [7]. Using proof of work, solving the hash puzzle is a difficult and time-consuming task. Without the right hardware, solving the hash could take even longer. This reduces the transaction processing rate, or the throughput.

Since the proof of stake algorithm spends less time doing complex computations, it can process far more transactions every second.

Fast finality times often go hand in hand with high throughput requirements. For example, processing credit card transactions also requires high throughput.

### 2.4. Governance

#### 2.4.1. Fork Resistance

Forks are bad in a consensus network. They essentially create multiple sources of truths that counter the establishment of trust in transactions recorded on a blockchain. Forks also create opportunities for double-spend attacks since a different version of the truth can be recorded in each fork.

Some consensus algorithms cannot avoid forking, at least temporarily, due to network latency and miner behavior. Neudecker and Hartenstein [9] have empirically analyzed forking in the Bitcoin network and concluded that the probability of a block to become part of the main chain increases linearly from its creation. This time window creates an opportunity for double spend.

While the issues created by this type of temporary forks can be mitigated by waiting for a sufficiently large number of subsequent blocks, it is the hard forks that cause major problems in the user community. Hard forks can rewrite the blockchain and make previously valid blocks invalid, or vice versa.

A good consensus protocol should have built-in mechanisms to detect as well as to deter hard forks.

### 2.4.2. Software Upgrades

Another aspect of the governance of a consensus algorithm is how the algorithm itself is updated. The algorithm needs to be updated from time to time to allow for critical security-related changes, fixing software bugs, adding more features, or for performance improvements.

Ensuring that the consensus algorithm can be leveraged for these critical software upgrade decisions is important for the viability of that blockchain.

## 2.5. Compliance

The first generation of consensus protocols was perceived as a way to get around the governmental regulatory and compliance issues by providing pseudonymity and confidentiality. Today, as governments across the world start looking into regulations for blockchain usage, it is important that the next generation of consensus protocols look at compliance-related features of the blockchain.

### 2.5.1. Regulatory compliance (e.g., KYC and AML)

Legal financial applications of the blockchain will subject the blockchain networks to a similar level of scrutiny as any other financial institution. Two primary regulatory requirements are likely to become important here.

Know Your Customer (KYC) requires financial institutions to verify certain aspects of a user. The protocol should ensure that it can exert some control over user accounts to allow for this.

Anti-Money Laundering (AML) and other fraud detection systems require the ability to monitor all transactions on the network and query history of transactions. Consensus protocols should have features to transparently distribute such information.

### 2.5.2. Removal of illegal content

Blockchains are built on the premise of immutability, and therefore it becomes impossible for blockchains to remove content. There are several instances of storage of illegal content (such as stolen classified documents) on public blockchains like the Bitcoin network. Short of a hard fork, removing this content violates the basic principle of immutability.

A good protocol should have a legitimate mechanism for the nodes to reach consensus to alter the blockchain in a controlled and transparent manner.

# 3. THE FRAMEWORK

With the backdrop of the discussion in the section above, Table 1 presents a framework to evaluate the consensus algorithms. The methodology used in determining the framework is as follows. The primary reason for using a blockchain in a business application is the trust and security that is ensured by decentralization. Hence, this model assigns over half of the weightage to security and decentralization requirements. However, we have seen that many initial

deployments of blockchain applications based on the early proof of work systems have suffered due to the inherent lack of scalability or governance models. Further, some governments have issued bans on certain blockchains due to lack of regulatory compliance requirements. This model emphasizes the importance of these requirements in today's blockchain systems by assigning nearly half the weightage to considerations related to scalability, governance and compliance. By balancing the security and decentralization requirements with those related to scalability, governance and compliance, this model achieves a holistic evaluation framework.

The weights provided in Table 1 are based on the analysis of the most common use cases of blockchain today. Based on the application that runs on the blockchain, the user can assign appropriate weights to each of the evaluation criteria. For some applications, scalability and time for finality may be much more important than resistance to double spending attacks (for example, it is inconceivable today to run all credit card transactions on the Bitcoin network because a user in front of a gas station will be unwilling to wait for several minutes for the block to be confirmed), while for other applications, security features might be far more important than the speed (for example, applications that register a deed for the ownership of a house on the blockchain). Based on a given application, the framework can be used to update the weights, and then a weighted score can be calculated to determine the best consensus algorithm for the given application.

Table 1. Comparative Analysis Framework

| Evaluation criteria | Description | Weightage |
|---|---|---|
| Security | | 25% |
| | Resistance to Sybil and Eclipse attacks | 7% |
| | Resistance to 51% attack | 7% |
| | Resistance to Internet-based attacks | 4% |
| | Resistance to Double Spend attacks | 7% |
| Decentralization | | 30% |
| | Decentralization through scale | 7% |
| | Geographical distribution of nodes | 7% |
| | Permissionless | 3% |
| | Open source | 7% |
| | Non requirement for specialized hardware | 6% |
| Scalability | | 20% |
| | Energy consumption | 7% |
| | Finality | 6% |
| | Throughput (Transactions per Second) | 7% |
| Governance | | 15% |
| | Fork resistance | 8% |
| | Software upgrades | 7% |
| Compliance | | 10% |
| | Regulatory compliance | 7% |
| | Removal of illegal content | 3% |

## 3.1. Limitations

As discussed above, the model presented above is generic in nature. While it is useful to select a blockchain consensus protocol for the vast majority of business and consumer applications, the model does not work for every application. There might be superseding considerations or

extenuating circumstances such as geographical data sovereignty requirements, or integrations with existing infrastructure that influence the decision of the consensus protocol. In such cases, the model presented above should be used in conjunction with other considerations.

## 4. FUTURE WORK

This paper presents a single evaluation framework, and as discussed in section 3.1, the presented framework may not be suitable for some niche use cases. The creation of multiple bespoke frameworks designed for a set of specific applications is recommended to address the needs of these use cases.

Currently, the framework is presented as a tool where the decision maker will enter scores against each criterion. We recommend the creation of an automated test-suite that can run against a target blockchain to evaluate and generate an automated score. An open source project that implements the test-suite could pave the way for objectively measuring effectiveness of the consensus protocols. Such a test tool can also inspire the design of a "perfect blockchain protocol" that can objectively maximize the score relative to the decision-making criteria described above.

## 5. CONCLUSIONS

In this paper, we discussed objective criteria for evaluating various consensus algorithms of a blockchain network. These criteria range from Security and Decentralization features, Scalability and Cost, Governance, and Compliance. We then presented a decision-making framework that holistically balances these criteria for a vast majority of business and consumer applications. This model can be used to make objective decisions about the selection of a consensus algorithm for blockchain based projects, as well as the comparison of new consensus algorithms against the existing ones.

### REFERENCES

[1] Ashar Ahmad, Abdulrahman Alabduljabbar, Muhammad Saad, DaeHun Nyang, Joongheon Kim, & David Mohaisen, (2021) "Empirically comparing the performance of blockchain's consensus algorithms", IET Blockchain Vol. 1, No. 1, pp. 56-64.

[2] Bin Cao, Zhenghui Zhang, Daquan Feng, Shengli Zhang, Lei Zhang, Mugen Peng, Yun Li (2020) "Performance analysis and comparison of PoW, PoS and DAG based blockchains", Digital Communications and Networks, Vol. 6, Issue 4, Nov 2020, pp480-485.

[3] Douceur, J.R. (2002) "The Sybil Attack", Revised Papers from the First International Workshop on Peer-to-Peer Systems, Springer: London, UK, pp. 251–260.

[4] S. S. Panda, B. K. Mohanta, U. Satapathy, D. Jena, D. Gountia and T. K. Patra, (2019) "Study of Blockchain Based Decentralized Consensus Algorithms," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), pp. 908-913.

[5] Sarwar Sayeed and Hector Marco-Gisbert (2019) "Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack", Applied Sciences, Vol. 9, Issue 9, Nov 2020, pp. 1788.

[6] Satoshi Nakamoto (2008) "Re: Bitcoin P2P e-cash paper", Available online at: https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html

[7] Seyed Bamakan, Amirhossein Motavali, & Alireza Bondarti, (2020) "A survey of blockchain consensus algorithms performance evaluation criteria", Expert Systems with Applications Vol. 154, pp. 113385.

[8] N. Chaudhry and M. M. Yousaf, (2018) "Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities," 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), pp. 54-63

[9] Till Neudecker and Hannes Hartenstein. (2019) "Short Paper: An Empirical Analysis of Blockchain Forks in Bitcoin", Lecture Notes in Computer Science book series (LNCS), Vol 11598, pp. 84-92.

## AUTHORS

Dipti Mahamuni is a senior year student at Arizona State University pursuing her BS in Computer Science from the Ira A. Fulton Schools of Engineering - School of Computing, Informatics, and Decision Systems Engineering.
Her areas of interest include Blockchain, AI/Machine learning and IoT.