Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 15, No. 2, 2024

Review of blockchain's consensus algorithms Comparative Analysis and Future Directions of Blockchain Consensus Mechanisms

Boyuan Yang

Jinan No.1 High school In Shandong Province, Jinan, China

Abstract: This paper provides a comprehensive review of blockchain consensus algorithms, classifying them into public chains, consortium chains, private chains, and special chains. It explores the evolution of these algorithms, from the early Nakamoto Consensus in Bitcoin to advanced mechanisms. The review highlights the varying degrees of decentralization, security, and scalability among these algorithms, addressing their advantages, limitations, and the ongoing research aimed at overcoming existing challenges. By comparing traditional and novel consensus mechanisms, this paper aims to provide a valuable reference for both new and seasoned researchers, fostering further innovation in the development of efficient and robust blockchain systems.

Keywords: Consensus algorithms; Public Chains; Consortium Chains; Private Chains; Blockchain.

1. Introduction

1.1. The history of blockchain

The history of blockchain technology traces back to the late 20th century. The concept of a cryptographically secured chain of blocks was first introduced in 1991 by researchers Stuart Haber and W. Scott Stornetta, who worked on a system where document timestamps could not be tampered with. However, it wasn't until 2008 that blockchain gained significant attention with the publication of the Bitcoin white paper by an anonymous person or group of people using the pseudonym Satoshi Nakamoto.

Nakamoto's paper, titled "Bitcoin: A Peer-to-Peer Electronic Cash System,"[1] proposed a decentralized digital currency that relied on a blockchain to record transactions. This blockchain was unique because it allowed for a distributed and secure method of recording transactions without the need for a trusted central authority. The first block of the Bitcoin blockchain, known as the "genesis block," was mined by Nakamoto in January 2009, marking the launch of the Bitcoin network.

Following the success of Bitcoin, blockchain technology began to attract the attention of developers, businesses, and researchers. In 2015, Ethereum was launched by Vitalik Buterin[2] and his team, introducing a new blockchain platform that supported smart contracts, self-executing contracts with the terms of the agreement directly written into code. This innovation expanded the potential applications of blockchain beyond digital currency to include decentralized applications (dApps) and various other use cases.

Since then, numerous blockchain projects have emerged, each aiming to address different challenges and improve upon the capabilities of existing blockchain systems. These projects include public blockchains like Bitcoin[1] and Ethereum[2], consortium blockchains for business collaborations, and private blockchains for specific organizational needs. The evolution of blockchain technology continues to unfold, with ongoing research and development focused on enhancing scalability, security, and interoperability.

1.2. Development of Blockchain consensus Algorithms

There are many consensus algorithms in blockchain, and the degree of decentralization, security and scalability of the algorithms are different. There is an urgent need to compare these algorithms. To provide a comprehensive understanding of the academic community. This can not only help new researchers quickly understand the frontier trends in this field, but also provide reference and reference for existing research. Finally, the advantages and disadvantages of the current blockchain consensus algorithm, as well as the existing research gaps and challenges are revealed. This can point the way for subsequent research and promote the birth of new and improved algorithms.

The domestic and foreign research status of blockchain algorithms shows the rapid development and diverse applications in this field. Traditional consensus algorithms such as PoW and PoS have been widely studied worldwide. International research has focused on improving the performance and efficiency of these algorithms, such as optimizing the energy consumption of PoW and improving the fairness of PoS. At the same time, researchers are also exploring new consensus mechanisms, such as Proof of Activity, Proof of Burn and Proof of Authority, to improve the speed of transactions and the security of the system.

2. Related Work

The history of blockchain consensus algorithms is a complex and diverse journey, covering the evolution from early basic theory to modern efficient algorithms. This journey not only shows the progress of technology, but also reflects the continuous exploration and innovation of human beings in distributed systems and security.

The Byzantine Generals Problem was first introduced by Leslie Lamport in 1982 and is the basis of the Byzantine Fault tolerance (BFT) theory[3]. The problem describes how to reach consensus in a distributed system in the presence of malicious nodes. The Byzantine Generals problem presents a fundamental theory of how to reach consensus in distributed systems in the presence of malicious nodes. In 1999, Miguel

Castro and Barbara Liskov proposed the Practical Byzantine Fault Tolerance (PBFT)[4], which was the first Byzantine fault tolerance algorithm applied in practice. PBFT allows the system to reach consensus even if at most one third of the nodes are malicious through a voting mechanism based on primary and backup nodes. The proposal of PBFT lays an important foundation for modern blockchain consensus algorithms. In 2008, Satoshi Nakamoto[5] published a white paper on Bitcoin that introduced the Nakamoto consensus, which stands for Proof of Work (PoW). Nakamoto consensus is a major breakthrough in blockchain technology, which realizes a decentralized and trustless consensus mechanism by means of mining competition. The success of Bitcoin has inspired the rise of the entire cryptocurrency and blockchain technology.

With the wide application of PoW, people gradually realize the problems of high energy consumption and low efficiency. Proof of Stake (PoS)[6] was first proposed as an alternative to PoW in 2011 by Sunny King and Scott Nadal in the Peercoin white paper. PoS selects verifiers by pledging tokens, which reduces energy consumption and improves efficiency. Cardano's Ouroboros[7] protocol further improved PoS in 2017, becoming the first PoS protocol to be rigorously academically validated. In 2014, Jae Kwon proposed Tendermint[8], an efficient consensus algorithm based on PBFT. Tendermint greatly improves the efficiency of the consensus process by simplifying the view change process[8], and becomes the core consensus mechanism of the Cosmos network[9]. Cosmos was launched in 2019 to enable cross-chain communication and interoperability. In 2016, Leemon Baird proposed Hashgraph Gossip about Gossip and Virtual Voting[10], a novel consensus algorithm based on Byzantine fault tolerance. Hashgraph achieves a high throughput and low latency consensus process through Gossip protocol and virtual voting mechanism. In 2017, Ethereum introduced Casper FFG, a hybrid consensus mechanism combining Nakamoto consensus and BFT designed to gradually transition from PoW to PoS. Casper FFG[11] lays the foundation for Ethereum 2.0 upgrade. In 2018, Maofan "Ted" Yin et al. proposed HotStuff[12], an improved consensus algorithm based on PBFT and Tendermint. HotStuff further improved consensus efficiency by simplifying the protocol state machine and became the core consensus algorithm for the Facebook Libra (now Diem) project. In 2018, Emin Gun Sirer et al. proposed the Avalanche family of protocols[13], which includes Slush, Snowflake, Snowball, and Avalanche[13]. Avalanche achieves a highly scalable and low-latency consensus process through random sampling and leaderless voting mechanism.In 2019, Spacemesh introduced Proof of Space-Time (PoST)[14], a consensus mechanism that combines the features of Nakamoto consensus and BFT. PoST enables efficient and decentralized consensus by leveraging a combination of disk space and time. In the same year, Celo introduced Lightweight Byzantine Fault Tolerance (LBFT)[15], an optimized consensus mechanism based on PBFT and other BFT protocols, with a focus on reducing communication complexity and improving efficiency. In 2020, LazyLedger[16] proposed an optimization based on Avalanche protocol, which used DAG (Directed Acyclic Graph) structure to improve parallel processing ability and throughput. In the same year, Polkadot launched Nominated Proof of Stake (NPoS)[17], based on Cardano's Ouroboros protocol, which selected the verifier through the nomination mechanism, improving the security and decentralization of the network.

In addition, There are some other consensus algorithms such as Proof of Activity (PoA)[18], Proof of Stake Velocity[19], Proof of Burn (PoB)[20], Scalable Byzantine Fault Tolerance[21], Proof of Authority[22], etc., which have innovated and optimized to different degrees in different application scenarios. For example, Proof of Activity combines the features of PoW and PoS, while Proof of Burn achieves consensus by destroying tokens.

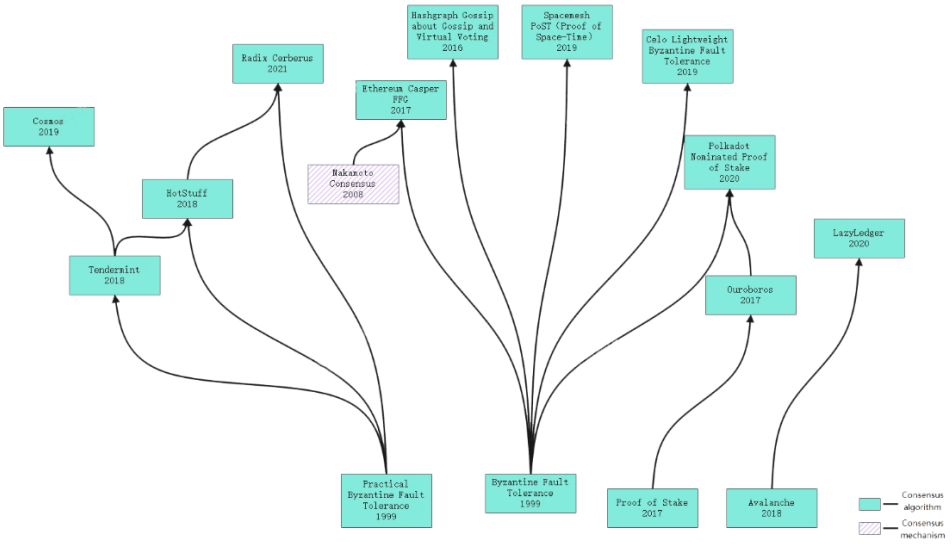

Figure 1. Summary of consensus algorithms

3. Classification of Consensus Algorithms

3.1. Public Chains

3.1.1. Radix Cerberus (2021)

Cerberus is a fragmented Byzantine Fault Tolerance (BFT)[23] consensus protocol for Radix platform, which improves system throughput and scalability by introducing partial sorting mechanism and state fragmentation. The core concepts include: first, the partial ordering mechanism allows the local ordering of related events rather than the global ordering, thus enabling the parallel processing of multiple

consensus instances. Secondly, state fragmentation divides the system state into multiple shards, and each shard runs as an independent BFT instance, which significantly reduces the bottleneck of global synchronization. Finally, Radix Engine, as an "asset-oriented" application layer, is able to express transactions in the form of discrete finite state machines, providing flexible application development and deployment capabilities. These designs ensure safety in asynchronous environments and liveness in synchronous environments, while achieving global consistency and efficient parallel consensus across shards through coordination of local Cerberus instances and newborn Cerberus instances.

3.1.2. Polkadot Nominated Proof of Stake

Polkadot's Nominated Proof of Stake (NPoS)[17],[19] is a unique consensus mechanism that aims to improve the security and decentralization of the network by introducing the roles of nominees and verifiers. The core concepts of NPoS include: First, verifiers are nodes responsible for verifying blocks and maintaining blockchain security. They are eligible for nomination and election by pledging a certain amount of DOT (Polkadot's native token); Second, nominators are participants who hold DOT and pledge it to support a particular verifier, and by selecting trustworthy verifiers, they help ensure the overall security of the network; Finally, through this mechanism, Polkadot realizes the interest binding between the verifier and the nominator, enhances the ability of the system to resist malicious behaviors, and promotes the decentralization and fairness of the network. This design allows validators and nominees to jointly participate in network governance and security maintenance, which ensures the stable and efficient operation of the entire Polkadot ecosystem.

3.1.3. Spacemesh PoST

Spacemesh's Proof of Space-Time (PoST)[14] is a consensus mechanism designed to secure a decentralized, permissionless ledger. It involves miners committing a specific amount of storage (space) for a set period (time), thus creating a verifiable resource called spacetime. The process consists of two phases: an initialization phase where miners commit data to the allocated space, and an execution phase where miners periodically prove they still possess the data. This ensures the miner has continuously used their allocated space over time. The system includes mechanisms to adjust for the cost of storage versus computational recomputation, maintaining the incentive for miners to store rather than recompute data. To enhance verification, Spacemesh employs a Proof of Elapsed Time (PoET) which, combined with PoST, forms a Non-Interactive Proof of Space-Time (NIPoST), ensuring a miner's commitment of spacetime over a given period

3.2. Consortium Chains

3.2.1. Lightweight Byzantine Fault Tolerance

Lightweight Byzantine Fault Tolerance (LBFT)[15] is an efficient consensus algorithm designed for the Industrial Internet of Things (IIoT) that optimizes the Practical Byzantine Fault Tolerance (PBFT) model. This algorithm introduces a credit score system to evaluate the reliability of nodes, ensuring that only well-performing nodes participate in the consensus process, while excluding or resetting nodes that exhibit Byzantine faults or frequent downtimes. It employs a dual-layer architecture where the lower layer verifies the legality of client transactions and the upper layer generates blocks and reaches consensus, synchronizing blocks to the lower layer for storage. Additionally, it features an adaptive master node selection algorithm that randomly selects nodes with high credit scores as master nodes, enhancing the efficiency of view changes and reducing the risk and resource consumption associated with malicious nodes becoming master nodes

3.2.2. HotStuff

HotStuff is a leader-based Byzantine fault-tolerant (BFT) consensus protocol designed for partially synchronous systems[12]. It achieves two key properties: linearity and responsiveness. Once network communication becomes synchronous, HotStuff allows a correct leader to drive consensus at the actual pace of network delay, rather than the maximum possible delay. This makes HotStuff both efficient and adaptable to real-world conditions. It uses a three-phase core to simplify leader replacement and consensus processes, reducing the complexity typically associated with view changes. By separating safety and liveness mechanisms, HotStuff also supports frequent leader rotations, enhancing its robustness and scalability for large-scale replication services

3.2.3. Scalable Byzantine Fault Tolerance

Scalable Byzantine Fault Tolerance (SBFT) is designed to enhance the efficiency and scalability of traditional Byzantine Fault Tolerance (BFT) protocols[21], particularly for large-scale distributed systems such as blockchains. SBFT aims to reduce communication complexity and improve performance by introducing optimizations like a stable leader protocol that supports optimistically linear, one round-trip decisions and an $O(n^2)$ communication view-change protocol. These improvements are achieved through methods such as a collector-based communication paradigm and the use of threshold cryptography for combining protocol votes. As a result, SBFT is capable of maintaining high throughput and low latency even in environments with a large number of nodes, addressing the scaling challenges inherent in traditional BFT solutions

3.3. Private Chain

3.3.1. Proof of Authority

Proof of Authority (PoA) is a consensus mechanism used in blockchain networks where a limited number of validators are pre-approved and known entities, typically chosen based on their trustworthiness and reputation. Validators in PoA systems are responsible for creating new blocks and validating transactions. This approach enhances the efficiency and speed of transaction processing, as the network doesn't need to expend significant computational resources to solve complex cryptographic puzzles like in Proof of Work (PoW). However, PoA relies on the reputation and integrity of validators, making it suitable for private or consortium blockchains where the participants are known and trusted. This centralized aspect of PoA ensures faster transaction times and higher throughput but introduces potential risks related to validator collusion or abuse of power.

3.4. Special Chains

3.4.1. LazyLedger

LazyLedger is a blockchain protocol designed for data availability and consensus[16]. It separates consensus from data availability by using a data availability layer that allows light clients to verify the availability of data without downloading the entire blockchain. This is achieved through the use of data availability proofs and fraud proofs, which ensure that all data needed for validating transactions is

available. LazyLedger's approach allows for scalable and efficient data storage, reducing the overhead for nodes and enabling high throughput and low-latency consensus mechanisms.

4. Parameter

4.1. Decentralization

The capacity of a blockchain system to continue functioning correctly even when there is a partition or network split that prevents some nodes from communicating with others. Partition tolerance is vital for ensuring that the blockchain remains operational even in the presence of network failures or disruptions, maintaining its robustness and reliability.

4.2. Scalability

The capacity of a blockchain network to handle an increasing number of transactions and nodes without compromising performance. Scalability is crucial for widespread adoption, ensuring that the blockchain can support a growing user base and high transaction throughput without degradation in speed or efficiency.

4.3. Security

The measures and mechanisms in place to protect the blockchain from attacks, fraud, and unauthorized access. High security is essential to maintaining the integrity, confidentiality, and availability of the blockchain data, protecting users and transactions from malicious activities.

4.4. Consistency

The guarantee that all nodes in the blockchain network have the same view of the data at all times. Consistency ensures that all participants agree on the state of the blockchain, which is fundamental for maintaining trust and accuracy in the system.

4.5. Availability

The ability of the blockchain network to remain operational and accessible for transactions and interactions at all times. High availability ensures that the blockchain can be used continuously without interruptions, making it reliable for users who depend on it for various applications.

4.6. Partition Tolerance

The capacity of a blockchain system to continue functioning correctly even when there is a partition or network split that prevents some nodes from communicating with others. Partition tolerance is vital for ensuring that the blockchain remains operational even in the presence of network failures or disruptions, maintaining its robustness and reliability.

Please refer to Table 1 for the specific scores.

Table 1. Specific scores about 8 Algorithms

Classification of Algorithms / Evaluation index	*Public Chains*			*Consortium Chains*			*Private Chain*	*Special Chains*
	Radix Cerberus	**NPoS**	**PoST**	**LBFT**	**HotStuff**	**SBFT**	**Proof of Authority**	**LazyLedger**
Decentralization	High	High	Medium	Medium	High	Medium	Low	High
Scalability	High	Medium	Medium	Medium	Medium	High	High	High
Security	Medium	Medium	Medium	Medium	High	Medium	Medium	Medium
Consistency	Medium	Medium	Medium	Medium	High	Medium	High	Medium
Availability	Medium	High	Medium	Medium	High	Medium	High	High
Partition Tolerance	Medium	Medium	Medium	Medium	Low	Medium	Low	Medium

5. Conclusion

This paper reviewed and compared various blockchain consensus algorithms, highlighting their evolution and performance in decentralization, security, and scalability. While traditional algorithms like PoW and PoS have foundational importance, newer mechanisms such as PoST, NPoS, and LBFT offer improved efficiency and performance. Despite progress, challenges remain, particularly in balancing these attributes. Future research should aim to optimize existing algorithms, explore hybrid models, and develop innovative solutions to meet the dynamic demands of blockchain applications. Continuous innovation in consensus algorithms is essential for advancing secure, scalable, and efficient blockchain systems, paving the way for broader adoption and new applications.

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[2] Buterin V. Ethereum white paper[J]. GitHub repository, 2013, 1: 22-23.

[3] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems (TOPLAS), 1982.

[4] Miguel Castro and Barbara Liskov. "Practical Byzantine Fault Tolerance." OSDI, 1999.

[5] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System." Bitcoin Whitepaper, 2008.

[6] Sunny King and Scott Nadal. "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake." Peercoin Whitepaper, 2012.

[7] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol." CRYPTO, 2017.

[8] Jae Kwon. "Tendermint: Consensus without Mining." Tendermint Whitepaper, 2014.

[9] Jae Kwon and Ethan Buchman. "Cosmos: A Network of Distributed Ledgers." Cosmos Whitepaper, 2019.

[10] Leemon Baird. "Hashgraph Consensus: Fair, Fast, Byzantine Fault Tolerance." Swirlds Technical Report, 2016.

[11] Vitalik Buterin and Virgil Griffith. "Casper the Friendly Finality Gadget." Ethereum Research Paper, 2017.

[12] Maofan "Ted" Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. "HotStuff: BFT Consensus in the Lens of Blockchain." PODC, 2019.

[13] Team Rocket. "Snowflake to Avalanche: A Novel Metastable Consensus Protocol Family for Cryptocurrencies." Avalanche Whitepaper, 2018.

[14] Tal Moran and Iddo Bentov. "PoST: Proofs of Space-Time." Spacemesh Protocol ePrint, 2019.

[15] Marek Olszewski, Sep Kamvar, Rene Reinsberg, and Marek Olszewski. "Celo: A Multi-Asset Cryptographic Protocol for Decentralized Social Payments." Celo Whitepaper, 2019.

[16] Mustafa Al-Bassam, Alberto Sonnino, and Vitalik Buterin. "LazyLedger: A Distributed Data Availability Ledger with Client-Controllable Scalability." LazyLedger Whitepaper, 2020.

[17] Gavin Wood. "Polkadot: Vision for a Heterogeneous Multi-Chain Framework." Polkadot Whitepaper, 2020.

[18] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake." ACM SIGMETRICS Performance Evaluation Review, 2014.

[19] Doug Pike and Steven Goldfeder. "PoSV: Proof of Stake Velocity: Building the Social Currency of the Digital Age." Reddcoin Whitepaper, 2014.

[20] Iain Stewart. "Proof of Burn." Bitcoin Wiki, 2012.

[21] Aleksandar Kuzmanovic, Lorenzo Alvisi, and Dahlia Malkhi. "SBFT: Scalable Byzantine Fault Tolerance." Arxiv Preprint, 2017.

[22] Gavin Wood. "Ethereum: A Secure Decentralized Generalized Transaction Ledger: Byzantium Version." Ethereum Yellow Paper, 2017.

[23] Cäsar F, Hughes D P, Primero J, et al. A Parallelized BFT Consensus Protocol for Radix[J]. 2020.