Research on blockchain consensus mechanism and implementation

Xingxiong Zhu\textsuperscript{1,2*}
\textsuperscript{1} State Grid Power Finance and E-Commerce Laboratory, 100053 Beijing, China
\textsuperscript{2} State Grid Electronic Commerce Co., Ltd., 100053 Beijing, China
\E-mail: zhuxx@pku.org.cn

Abstract. Each of the most popular blockchain platform, Bitcoin, Ethereum, and Hyperledger Fabric, varies in aspects of decentralization, permission, anonymity, and native-currency, has its own consensus mechanism, algorithm and implementation. In the mainstream blockchain technology, there are many common consensus algorithms. They differ in terms of computational complexity, fault-tolerance, and resilience. The performance, consistency, scalability, and efficiency of blockchain consensus mechanism need further improvement and optimization. Consensus mechanism and code implementation of Bitcoin, Ethereum and Hyperledger are analyzed, discussed and proposed.

1. Introduction
Each of the most popular blockchain platform, Bitcoin, Ethereum, and Hyperledger Fabric, varies in aspects of decentralization, permission, anonymity, and native-currency, has its own consensus mechanism, algorithm and implementation.

In the mainstream blockchain technology, there are many common consensus algorithms, such as proof-of-work (PoW), proof-of-stake (PoS), practical byzantine fault-tolerance (PBFT), delegated proof-of-stake (DPOS), Ripple, and Tendermint. They differ in terms of computational complexity, fault-tolerance, scalability, performance, and effectiveness.

2. Consensus mechanism
Consensus means to reach agreements among network nodes or systems, according to the interest of the system participants. Consensus algorithm is the method for achieving information consistent. The topology of network points is a directed graph $G$, the formulas [1] are as follow, where, $V$ is points set, $E$ is lines set, the near points of $i$ are $N_i$, and the consensus is between point $i$ and all of its near points $N_i$.

\begin{align}
V &= \{1, 2, ..., n\} \\
E &\subseteq V \times V \\
G &= (V, E) \\
N_i &= \{j \in V : (i, j) \in E\}
\end{align}

3. Bitcoin consensus
Bitcoin consensus mechanism and implementation are analyzed in the following.
3.1. Bitcoin consensus mechanism
The consensus mechanism of Bitcoin is proof-of-work [2] that nodes accept of valid blocks by increasing them.

To add new block to the chain, the node has to execute calculate work, known as PoW. It needs to obtain a hash value, less than a certain value [3]. The winner who finds the hash value that meets the requirement, adds the proposal block to the chain, and obtains the reward.

The Bitcoin consensus is about transaction rules, transaction states, and Bitcoin values [4]. It is agreement of rules that determinate which blocks and transactions are valid or not, agreement of which transactions have occurred, and agreement of that bitcoins have value and players want accept bitcoins in payment.

Hashcash proof-of-work is utilized, using SHA-256. A hash of the block header, not exceeding a certain value, is created to prove the computation work. It only uses the 80-byte block header to hash.

3.2. Implementation of Bitcoin consensus
The Bitcoin consensus source code [5] includes key files, such as consensus header, merkle, validation, parameters, transaction check, and transaction verify. The pow files implement proof-of-work, as figure 1 shows.

4. Ethereum consensus
Ethereum consensus mechanism and source code implementation are discussed in the following.

4.1. Ethereum consensus mechanism
The initial and current version Ethereum uses PoW as the consensus mechanism, where participants work to solve mathematics puzzles.

The GHOST protocol [6], Greedy Heaviest Observed Subtree, was once utilized in Ethereum. From root, the genesis block, to the leaf, containing the most recent transaction, the heaviest path has the most computation done among it. The difficulty is recorded in the block header. The total difficulty of all blocks of a chain validates the total computation done. GHOST protocol uses heaviest subtree rule replace Bitcoin’s longest chain.
Proof-of-stake consensus algorithm, named Casper, is planned to be used in the future, because of PoW massive resource expenditure. PoS depends on the validator’s economic stake, the weight of each validator’s vote is according to its deposit, which is stake.

4.2. Implementation of Ethereum consensus

The source code of Ethereum [7] consensus is depicted in Fig.2. Ethash is the PoW module for Ethereum.

![Figure 2. Implementation of Ethereum consensus](image)

5. Hyperledger Fabric consensus

Hyperledger Fabric consensus mechanism and source code implementation are proposed in the following.

5.1. Hyperledger consensus mechanism

Hyperledger Fabric uses Practical Byzantine Fault-Tolerance as its consensus protocol. In an independent nodes system, it tolerates $f < n/3$ nodes [8]. With no more than $f$ faulty nodes, such as error, crash, shutdown, and subverted, the other $n - f$ nodes are on service correctly.

A Validating Peer (VP) represents a participating institute, and one leader is elected. A transaction request is made by the client to their respective VP, which validates and broadcasts the transaction to other VPs. A block, including the pending transactions, is created by the leader peer. The candidate block is broadcasted for consensus using PBFT [9]. If at least $2f+1$ peers agree, all transactions in this block are executed by each VP, and the new block is appended to each VP ledger.

5.2. Implementation of Hyperledger consensus

The source code of Hyperledger Fabric [10] gossip consensus is depicted in Fig.3. Two phases, push and pull, are used for information dissemination among peers. Blocks are distributed to peers through gossip or ordering service. The peak mint or spend throughput is higher with gossip service [11].
6. Discussion
In order to improve modularity and reusability, each consensus algorithm, such as PoW, PoS, PBFT, DPOS, Ripple, and Tendermint, should be packaged into a library. It is separated and extracted from the platform, and reused by wide variety of systems.

Different consensus mechanisms are combined, with convergence and application of a variety of consensus algorithms to enhance strengths and avoid weaknesses, improve performance and efficiency.

7. Conclusion
In the mainstream blockchain technology, there are many common consensus algorithms. They differ in terms of computational complexity, fault-tolerance, and resilience. The performance, consistency, scalability, and efficiency of blockchain consensus mechanism need further improvement and optimization.

Consensus mechanism and code implementation of Bitcoin, Ethereum and Hyperledger are analyzed, discussed and proposed.

Acknowledgment
This work was supported by grants from the science and technology project of State Grid Corporation of China (No. 52110417001D).

Author introduction
Xingxiong Zhu received the master degree in software engineering from Peking University, China in 2005. He was one of the main drafters of China National Standard GB/T 25656-2010, Information Technology Specification for Chinese Linux Programming Interface. Email, zhxux@pku.org.cn.

References
[1] Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1), 215-233.
[2] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.
[3] Baliga, A. (2017). Understanding blockchain consensus models. In Persistent.
[4] Kroll, J. A., Davey, I. C., & Felten, E. W. (2013, June). The economics of Bitcoin mining, or
Bitcoin in the presence of adversaries. In *Proceedings of WEIS* (Vol. 2013, p. 11).

[5] The Bitcoin Core developers. Bitcoin Core source code. https://github.com/bitcoin/bitcoin.

[6] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151, 1-32.

[7] The go-ethereum Authors. Go Ethereum. https://github.com/ethereum/go-ethereum.

[8] Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*.

[9] Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., & Rindos, A. (2017, September). Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (pp. 253-255). IEEE.

[10] Hyperledger Fabric v1.4 Project. Hyperledger Fabric source code. https://github.com/hyperledger/fabric.

[11] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Muralidharan, S. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (p. 30). ACM.