Available online at www.sciencedirect.com

ScienceDirect

ICT Express 6 (2020) 93–97

www.elsevier.com/locate/icte

Analysis of the main consensus protocols of blockchain

Shijie Zhang[a], Jong-Hyouk Lee[b,*]

[a] *Protocol Engineering Lab., Sangmyung University, Republic of Korea*
[b] *Sejong University, Republic of Korea*

Received 15 May 2019; accepted 7 August 2019
Available online 12 August 2019

Abstract

Blockchain is the core technology of many cryptocurrencies. Blockchain as a distributed ledger technology has received extensive research attention. In addition to cryptography and P2P (peer-to-peer) technology, consensus protocols are also a fundamental part of the blockchain technology. A good consensus protocol can guarantee the fault tolerance and security of the blockchain systems. The consensus protocols currently used in most blockchain systems can be broadly divided into two categories: the probabilistic-finality consensus protocols and the absolute-finality consensus protocols. This paper introduces some of the main consensus protocols of these two categories, and analyzes their strengths and weaknesses as well as the applicable blockchain types.

Keywords: Blockchains; Consensus protocols; Cryptocurrencies

1. Introduction

Blockchain first appeared in Nakamoto's Bitcoin white paper that describes a new decentralized cryptocurrency [1]. Bitcoin takes the blockchain technology to the extreme and attracts people's wide attention. Afterward, many cryptocurrencies and projects based on the blockchain spring up. Blockchain thus has become a hot topic. Interestingly, the technology adopted by the blockchain is not new. Blockchain simply combines cryptography, distributed system technology, peer-to-peer networking technology and other well-known technologies. Besides, blockchain also provides a secure framework for the cryptocurrencies, in which anyone cannot tamper the content of transactions and all the nodes participate in transactions anonymously. For this reason, the blockchain technology can be widely used in various fields, e.g., financial field, medical systems, supply chain, and Internet of Things (IoT).

However, in the process of applying the blockchain technology, there will be many challenges and issues, among which how to design an appropriate consensus protocol is a big issue. The consensus of blockchain is that all nodes maintain the same distributed ledger. In traditional software architecture, the consensus is hardly a problem due to the existence of the center server, hence the other nodes only need to be aligned with the server. However, in a distributed network such as blockchain, each node is both a host and a server, and it needs to exchange information with other nodes to reach a consensus. Sometimes some nodes will be down or offline, and there will also be some malicious nodes, which will seriously affect or destroy the process of consensus. Therefore, an excellent consensus protocol can tolerate the occurrence of these phenomena and minimize the harm so as not to affect the final consensus result. In addition, the consensus protocol adopted by the system also needs to be suitable for the blockchain type used by the system. There are three basic types of blockchain: public blockchain, consortium blockchain and private blockchain [2]. Each type of blockchain has different application scenarios. The adopted consensus protocol thus needs to fit the demands of specific application scenario. In this paper, we introduce some main consensus protocols of blockchain and analyze their performance and application scenarios.

2. Main consensus protocols

In distributed systems, there is no perfect consensus protocol. The consensus protocol needs to make a trade-off

[*] Corresponding author.
E-mail address: jonghyouk@sejong.ac.kr (J.-H. Lee).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2019.08.001
2405-9595/© 2020 The Korean Institute of Communications and Information Sciences (KICS). Publishing services by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

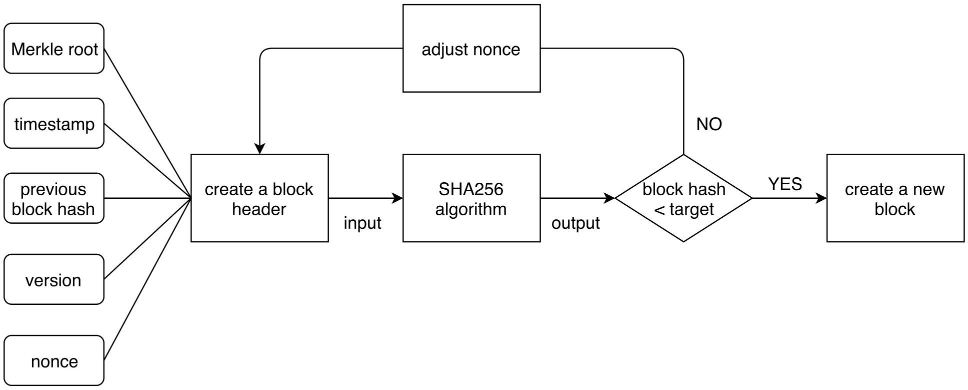

Fig. 1. Flow of PoW.

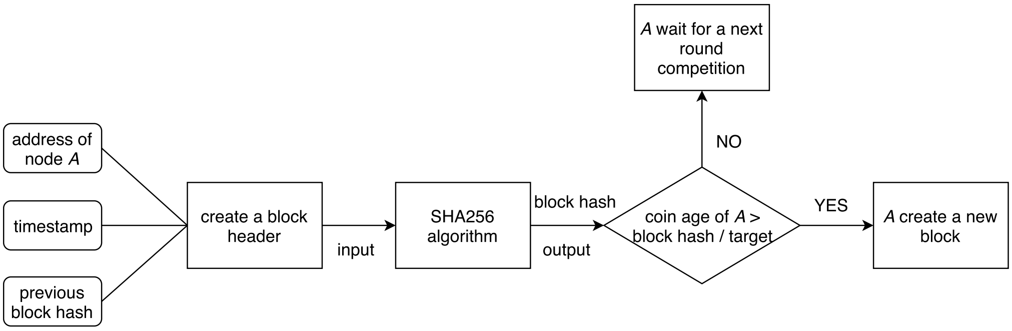

Fig. 2. Flow of PoS.

among consistency, availability and partition fault tolerance (CAP) [3]. Besides, the consensus protocol also needs to address Byzantine Generals Problem that there will be some malicious nodes deliberately undermining the consensus process [4]. In this section, we make a detailed description of some popular blockchain consensus protocols that can effectively address Byzantine Generals Problem.

PoW (Proof of Work): PoW is adopted by Bitcoin, Ethereum, etc [1,5]. PoW selects one node to create a new block in each round of consensus by computational power competition. In the competition, the participating nodes need to solve a cryptographic puzzle. The node who first addresses the puzzle can have a right to create a new block. The flow of the block creation in PoW is presented in Fig. 1. It is very difficult to solve a PoW puzzle. Nodes need to keep adjusting the value of nonce to get the correct answer, which requires much computational power. It is feasible for a malicious attacker to overthrow one block in a chain, but as the valid blocks in the chain increase, the workload is also accumulated, therefore overthrowing a long chain requires a huge amount of computational power. PoW belongs to the probabilistic-finality consensus protocols since it guarantees eventual consistency.

PoS (Proof of Stake): In PoS, selecting each round of node who creates a new block depends on the held stake rather than the computational power. Although nodes still need to solve a SHA256 puzzle:

$$SHA256(timestamp,\ previous\ hash \ldots) < target \times coin.$$

The different from PoW is that nodes do not need to adjust nonce for many times, instead, the key to solve this puzzle is the amount of stake (coins). Hence, PoS is an energy-saving consensus protocol, which leverages a way of the internal currency incentive instead of consuming lots of computational power to reach a consensus. The flow of PoS is shown in Fig. 2. Like PoW, PoS is also a probabilistic-finality consensus protocol. PPcoin was the first cryptocurrency to apply PoS to the blockchain. In PPcoin, in addition to the size of the stake, the coin age is also introduced in solving a PoS puzzle [6]. For instance, if you hold 10 coins for a total of 20 days, then your coin age is 200. Once a node creates a new block, his coin age will be cleared to 0. In addition to PPcoin, many cryptocurrencies adopt PoS, e.g., Nxt, Ouroboros [7,8]. Note that Ethereum plans to transition from PoW to PoS.

DPoS (Delegated Proof of Stake): The principle of DPoS is to let nodes who hold stake vote to elect block verifiers (i.e., block creators) [9]. This way of voting makes the stakeholders give the right of creating blocks to the delegates they support instead of creating blocks themselves, thus reducing their computational power consumption to 0. We can clearly see the flow of DPoS in Fig. 3. DPoS is like a parliamentary system, as shown in Fig. 3, if the delegates are unable to generate blocks in their turns, they will be dismissed and the stakeholders will select new nodes to replace them. DPoS makes the most use of the shareholders' votes to reach a consensus in a fair and democratic way. Compared to PoW and PoS, DPoS is a low-cost and high-efficiency consensus protocol. There are also some cryptocurrencies adopting

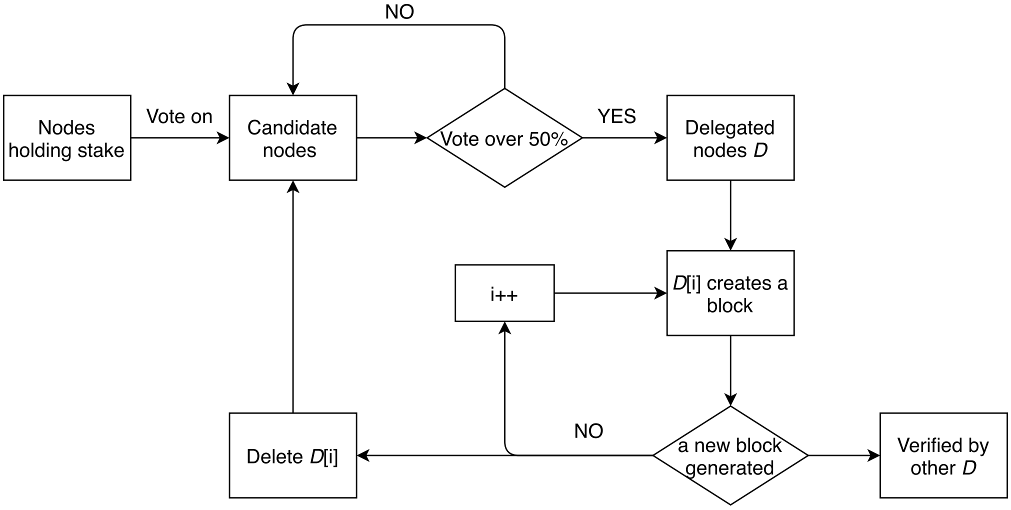

Fig. 3. Flow of DPoS.

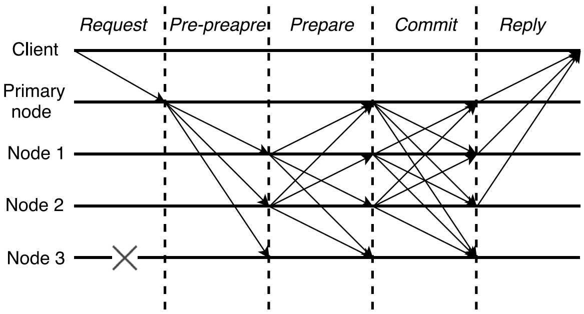

Fig. 4. Process of PBFT.

DPoS such as BitShares, EOS, etc [9,10]. The new version of EOS has turned DPoS to BFT-DPoS (Byzantine Fault Tolerance-DPoS) [10].

PBFT (Practical Byzantine Fault Tolerance): PBFT is a Byzantine Fault Tolerance protocol with low algorithm complexity and high practicality in distributed systems [11]. PBFT contains five phases: *request*, *pre-prepare*, *prepare*, *commit* and *reply*. Fig. 4 describes how PBFT works. The primary node forwards the message sent by the client to the other three nodes. In the case that node 3 is crashed, one message goes through five phases to reach a consensus among these nodes. Finally, these nodes reply to the client to complete a round of consensus. PBFT guarantees nodes maintain a common state and take a consistent action in each round of consensus. PBFT achieves the goal of strong consistency, thus it is an absolute-finality consensus protocol. As mentioned before, EOS takes a combined consensus protocol. EOS leverages PBFT to simultaneously work with the block validation and creation in DPoS, greatly reducing the time required for a round of consensus [10]. A new protocol called Stellar is an improvement of PBFT. Stellar adopts FBA (Federated Byzantine Agreement) protocol, in which nodes can choose the federation they trust to conduct the consensus process [12].

Ripple: Ripple is an open source payment protocol [13]. In Ripple, transactions are initiated by clients and broadcast throughout the network via tracking nodes or validating nodes. However, the consensus process in Ripple is performed by validating nodes, each of which owns a list of trusted nodes called UNL (Unique Node List). Nodes in UNL can vote on the transactions they support. The process of consensus in Ripple is presented in Fig. 5. Each validating node sends its own transactions set as a proposal to other validating nodes. Once receiving the transaction proposals sent by nodes in UNL, the validating node will check each transaction in the proposal. The transaction in the proposal will get one vote if there is the same transaction in its local transactions set. When the transaction gets more than 50% of the votes, this transaction will enter the next round. The screening threshold will be increased for each round, and transactions with more than 80% of the votes will be finally recorded in the distributed ledger. Hence, Ripple is an absolute-finality consensus protocol.

3. Analysis and comparison

In this section, we analyze the main consensus protocols mentioned in Section 2 in terms of fault tolerance, limitation, scalability, and application scenarios. The analysis and comparison results are summarized in Table 1.

3.1. Fault tolerance

PoW, PoS and DPoS are probabilistic-finality protocols, and attackers need to accumulate a large amount of computational power or coins (stake) to create a long private chain to replace a valid chain. For instance, in Bitcoin, a 50% fraction of the computational power is sufficient for an attacker to create a longer private chain to successfully complete a double-spend attack [1]. Hence, if attacker's fraction of the computational power is more than or equal to 50%, the blockchain network will be undermined. Like PoW, PoS and DPoS can only allow the existence of the stakeholder with less than 50% of the held stake. In PBFT, if there are a total of $3f + 1$ nodes in the

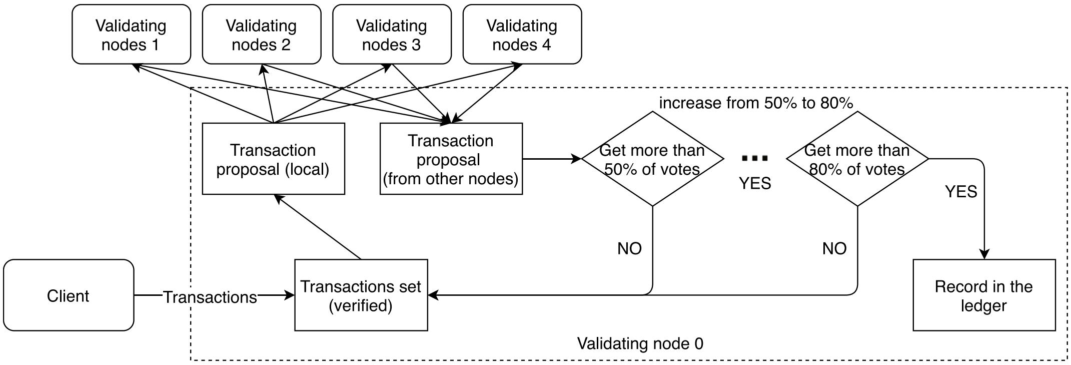

Fig. 5. Process of Ripple.

Table 1
Main consensus protocols comparison.

Property	PoW	PoS	DPoS	PBFT	Ripple
Type	Probabilistic-finality	Probabilistic-finality	Probabilistic-finality	Absolute-finality	Absolute-finality
Fault tolerance	50%	50%	50%	33%	20%
Power consumption	Large	Less	Less	Negligible	Negligible
Scalability	Good	Good	Good	Bad	Good
Application	Public	Public	Public	Permissioned	Permissioned

network, the number of normal nodes must exceed $2f + 1$, which means that the number of malicious or crashed nodes must be less than f. Therefore, the fault tolerance of PBFT is 1/3 [11]. The fault tolerance of Ripple is only 20%, i.e., Ripple can tolerate Byzantine Problem in 20% of nodes in the entire network without affecting the correct result of consensus [13].

3.2. Limitation

There is no doubt that PoW consumes the most computational power among these consensus protocols, and the transaction throughput per second (TPS) of Bitcoin adopting PoW is only 3–7, which greatly limits the application prospect of PoW in actual payment. PoS and DPoS have similar shortcomings, although they can greatly reduce the consumption of the computational power, only the stakeholders can get the block reward, which leads to the great reduction in the liquidity of coins in DPoS, the poorer the poor and the richer the rich. PBFT requires each node communicating with other nodes to exchange messages in each round of the consensus, PBFT thus has extremely high performance requirements for the network. Moreover, the identity of each node participating in the consensus is known, so there is no guarantee on the anonymity. In Ripple, a round of consensus process is completed in a few seconds, which is suitable for the actual payment scenario. However, Ripple is managed and controlled by a few organizations, which does not satisfy the decentralization nature of blockchain.

3.3. Scalability

PoW, PoS and DPoS all have good scalability. Although TPS of them is not very high, there are some ways that can help improve the scalability. For instances, Bitcoin adopts lightning network to provide an off-chain payment to improve the scalability [14]. Ethereum proposed the sharding technology and Plasma, which are layer 1 and layer 2 scaling solutions, respectively [15]. The scalability of PBFT is limited since PBFT is suitable for a high performance network with a small number of nodes. Unlike PBFT, Ripple can be suitable for a large scale network, and TPS of Ripple is over 1500, hence Ripple has strong scalability.

3.4. Scenarios

As mentioned before, current blockchain systems can be categorized into three types. In public blockchain, everyone can take part in the consensus process and the distributed ledger is visible to the public. PoW, PoS, and DPoS can be applied to public blockchain. Private blockchain and consortium blockchain belong to the permissioned blockchain as only permitted nodes can participate in the consensus process. The identity of each node is known to the public in PBFT and Ripple, thus they are all suitable for private blockchain or consortium blockchain. Although private blockchain and consortium blockchain are not as decentralized as public blockchain, due to the strong consistency and high efficiency of consensus, they are more suitable for some commercial and medical scenarios.

4. Conclusion

The consensus protocol is the guarantee for the stable operation of blockchain systems. Nodes agree on a certain value or transaction through the consensus protocol. In this paper, we introduced some popular blockchain consensus protocols and found their strengths, weaknesses and application scenarios through analysis and comparison. We concluded that designing a good consensus protocol should consider not only good fault tolerance but also how to make the best use of it in the appropriate application scenario.

Declaration of competing interest

The authors declare that there is no conflict of interest in this paper.

References

[1] S. Nakamoto, et al., Bitcoin: A peer-to-peer electronic cash system.

[2] V. Buterin, On public and private blockchains, https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/, 2015.

[3] S. Gilbert, N. Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT News 33 (2) (2002) 51–59.

[4] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, ACM Trans. Program. Lang. Syst. (TOPLAS) 4 (3) (1982) 382–401.

[5] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Proj. Yellow Pap. 151 (2014) 1–32.

[6] S. King, S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, self-published paper, August 19.

[7] N. community, Whitepaper:Nxt, http://nxtwiki.org/wiki/Whitepaper:Nxt.

[8] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: Annual International Cryptology Conference, Springer, 2017, pp. 357–388.

[9] D. Larimer, Delegated proof-of-stake (dpos), Bitshare whitepaper.

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

[11] M. Castro, B. Liskov, et al., Practical byzantine fault tolerance, in: OSDI, Vol. 99, 1999, pp. 173–186.

[12] D. Mazieres, The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus, Stellar Development Foundation, Citeseer, 2015.

[13] D. Schwartz, N. Youngs, A. Britto, et al., The ripple protocol consensus algorithm, Ripple Labs Inc White Paper 5.

[14] J. Poon, T. Dryja, The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[15] J. Poon, V. Buterin, Plasma: Scalable autonomous smart contracts, White paper, 1–47, 2017.