Wireless Networks (2020) 26:5579–5593
https://doi.org/10.1007/s11276-019-02195-0

Recent advances in consensus protocols for blockchain: a survey

Shaohua Wan[1,2] · Meijun Li[3] · Gaoyang Liu[3] · Chen Wang[3]

Published online: 25 November 2019

Abstract

As the core of a blockchain system, the consensus mechanism not only helps to maintain the consistency of node data, but also gets involved in the issuance of tokens and prevention of attacks. Since the first blockchain system was born, it has been continuously improved with the development of blockchain technology and evolved into multiple new branches. Starting with the basic introduction of consensus and the classic Byzantine Generals Problem in distributed computing area, this survey utilizes a thorough classification to explain current consensus protocols in the blockchain system, presents the characteristics of mainstream protocols (PoW, PoS, DPoS, PBFT, etc.) and analyzes the strengths and weaknesses of them. Then we evaluate the performance qualitatively and quantitatively. In the end, we highlight several research directions for developing more practical consensus protocols for the future.

Keywords Blockchain · Consensus protocol · PoW · PoS · DPoS · PBFT

1 Introduction

Born as the underlying technology of Bitcoin [40], a blockchain is a decentralized database which is comprised of a series of data blocks attached by cryptography. Each data block contains a batch of transactions to verify the validity of information and generate the next block. The blockchain absorbs miscellaneous techniques such as distributed architecture, peer-to-peer network protocol, encryption algorithm, smart contract, identity authentication, cloud computing, etc., becoming a transparent and highly-reliable overall technical solution [47]. It can be widely applied into scenarios such as mobile Internet [11, 36, 46, 51, 62], sensor networks [27, 53–56], as well as Internet of Things [2, 23, 35, 63, 66]. For example, blockchain-as-a-service [60] affords mobile content providers with an ecosystem that stores and controls their content across the entire mobile network, enabling copyright protection, auto online marketing and eliminating the risk of hacking and content redistribution. Blockchain can also help deploy 5G technologies and assist next-generation distributed wireless networks by providing seamless access across heterogeneous devices and multiple networks [1, 16, 24, 52].

Essentially, a blockchain system is an asynchronous distributed system that could be analyzed as a set of state machine replications (SMRs) [44]. Each blockchain node involved in recording data is an SMR, and the data it records is the current state. Appending a verified block to the system by each node is equivalent to an operation that changes the current state. To achieve a consistent state for all nodes in the system, it acquires the consistent initial state of each node, and the consistent operation adding to the system each time. This process/algorithm of achieving the consistency of distributed nodes is the consensus.

In a distributed system, the consistency problem is an important and classic problem studied since the 1970s. There is a basic assumption that nodes participating in the

✉ Chen Wang
chenwang@hust.edu.cn

Shaohua Wan
shaohua.wan@ieee.org

Meijun Li
meijunli@hust.edu.cn

Gaoyang Liu
liugaoyang@hust.edu.cn

[1] State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

[2] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

[3] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

calculation are not reliable and may fail. Normally, the failures come in two types: crash failures and Byzantine failures [48]. The difference is that the former only lose normal functions, while the latter not only work improperly but also could maliciously interfere with normal nodes' work. The term Byzantine failure is derived from the Byzantine Generals Problem described by Leslie Lamport [31]: Due to the vast territory of the Byzantine Roman Empire, each royal army is separated far apart for defense, and generals of different armies can only rely on the messengers to exchange information. Before each action, they must agree on whether to attack or retreat. But there could be traitors among all the generals, and they may send wrong messages intentionally to interfere with others. In that case, how can a loyal general unify his plan of war with the knowledge of a traitor? This is the Byzantine Generals Problem. In a blockchain system, each node can be seen as a general who wants to ensure the consistency of the blockchain ledger. However, there may be malicious nodes trying to tamper with the content of the ledger and obtain greater economic revenues. How to deal with the problem depends on the design and implementation of the consensus mechanism.

Consensus is the soul of blockchain. Motivated by the first consensus in Bitcoin, researchers have put forward many variants to endow blockchains with better performance. As more and more consensus protocols are proposed, it is of urgent need to give an explicit investigation and comparison to them. Zheng et al. [67] survey six kinds of consensus and compare them in node identity management, energy-saving and power of fault tolerance. In [10], they analyze recent permissioned consensus protocols from security performance and fault tolerance. Nguyen et al. [41] categorize the mainstream consensus protocols into two kinds: proof-based and vote-based consensus protocols and highlight the differences between these two kinds. Wang et al. [57] focus on the designing methodologies and permissionless algorithms to investigate their influence on blockchain applications.

Nevertheless, these surveys are deficient in various aspects of comparison, especially in consensus design and qualitative and quantitative performance. Our work is to summarize and analyze the recent advances in blockchain consensus protocols, by giving an explicit comparison of their performance and other critical particularities. This work features the following contributions: (a) reviewing the representative protocols according to consensus classification, (b) providing a comprehensive comparison of their qualitative and quantitative performance and other critical particularities with pros and cons, and (c) discussing future research trends in consensus studies.

The rest of this paper is organized as follows. Section 2 describes the classification of existing consensus protocols and the introduction of these protocols. The comparative performance analysis of the existing consensus protocols is done in Sect. 3. Section 4 points out future directions on the development of blockchain consensus. Finally, Sect. 5 draws the conclusion.

2 Consensus classification

Based on whether the number of nodes in the calculation is certain or not, and whether the nodes have malicious behaviors, we can divide the current consensus protocols into four cases. Considering the assumption that the number of nodes is uncertain and the nodes do not have malicious behaviors is too ideal, we only talk about the other three practical cases shown in Fig. 1. Note that in practice the blockchain is more considered about the consensus among untrusted nodes, so the "non-BFT (Byzantine Fault Tolerance) consensus with limited nodes" in the dashed box in Fig. 1 only exists in theory. Moreover, such kind of blockchain systems, as far as we know, has not yet emerged. Even if such systems exist, they are only suitable for highly trusted private networks. In the following subsections, we will introduce different consensus protocols mentioned in this classification.

2.1 PoW

The Proof-of-Work (PoW) consensus is the first and most widely-adopted consensus protocol in current blockchain systems. Simply speaking, PoW is proof to confirm that you have done a certain amount of work. Its concept was first proposed by Cynthia Dwork and Moni Naor in 1993 to resolve the problem of spam mail [21]. The basic idea is that, before sending a mail message, the user is required to send a proof-of-work related to the message. This proof-of-work is usually a process which aims to solve a mathematical problem and the problem should meet the following conditions:

- Be related to the messages to defend replay attacks against PoW.
- Be difficult enough to prevent being cracked by the third party.
- Be easy enough to verify the recipient, so as to avoid excessive computing overhead.

In [3], another anti-spam system used PoW for Hashcash. After that, Nakamoto adopted this innovative mechanism to achieve the consistency of nodes in Bitcoin in 2008 [40], laying a foundation for various blockchains and consensus protocols in others' later work.

For the Bitcoin network, Nakamoto has improved the traditional PoW consensus. To distinguish it from the

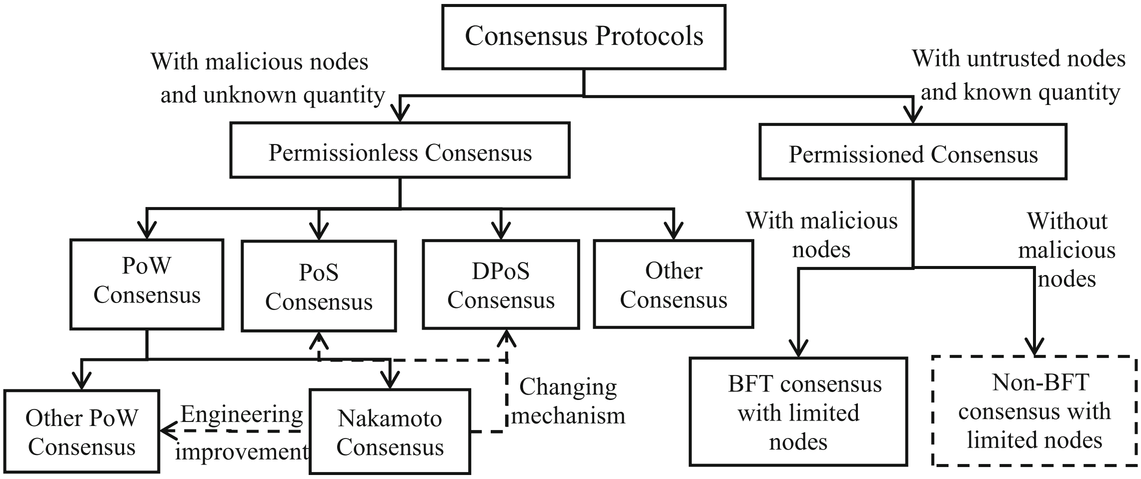


Fig. 1 Consensus protocols in blockchain systems

earlier one, we call it Nakamoto Consensus here. The mathematical puzzle that Nakamoto Consensus adopted is to solve a 256-bit integer *Nonce* as a so-called lucky number, ensuring the hash value of it and the created block header is less than a "difficulty", i.e.,

$$H(B) \leq m \tag{1}$$

Here, B is the block to submit. H is a hash function and m is the difficulty, a very small real number determined by the nature of the hash function.

If a required *Nonce* is found and approved in the Bitcoin system, the discoverer can receive a corresponding amount of Bitcoins as a reward. Because violently seeking *Nonce* requires a lot of calculations, the process of calculating is thus vividly called "mining". In order to adapt to the dynamic changes of the computing power of the entire system, it is ensured that the system generates blocks roughly at a predetermined rate (about one block per 10 minutes). The difficulty is dynamically adjustable, and the adjustment is also based on the consensus. The adjustment period is approximately one week (i.e., adjust the difficulty per $24 \times 6 \times 7$ blocks). In the PoW mechanism, since the expected time to find the *Nonce* can be adjusted, a mechanism of decentralized time series is constructed. At the same time, the decision problem of multiple decentralized nodes is also solved, that is, the entire network uses the data submitted by the node that first finds the legal *Nonce*.

Next, let us see how to reach the consensus. After any honest node generates a new block, it broadcasts the block to the entire network. For other honest nodes, they verify the correctness of the newly-received block. If the block is proved to be valid, they will abandon their ongoing block calculations, then reselect the transaction not added to the blockchain from the received list of transactions based on the received new block, generate a new block header and perform a new round of *Nonce* calculation.

Since the transactions received by different nodes have precedence, it may cause one node to receive two or more legitimate blocks, which leads to a temporary fork, like Fig. 2(a). After the fork occurs, each node can only continue to generate new blocks based on one of the new blocks, until one of the forks wins the competition. The fork is only temporary; as the time grows, it will be replaced by the longest chain, as Fig. 2(b) shows. Once a blockchain node decides to generate a new block based on a certain block, it means that the node permits the block and all other previous records. This permission is based on probability. If the chains published by other nodes are longer, the node will abandon the former consensus. Although the consensus is based on probability, it can be proved when the total computing power of the nodes participating in block generation is not dominant, that is, when the computing power is lower than 51% of the total computing power in the entire network [40], the probability that the nth block before the current block is discarded is exponentially negatively correlated with n, i.e., the larger n is, the lower the probability of the nth preceding node in

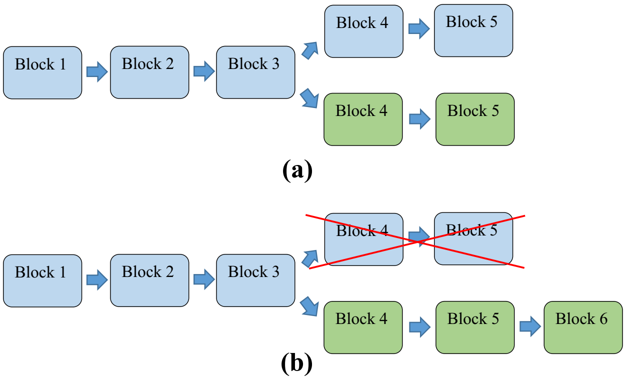


Fig. 2 The fork forms and disappears in the PoW Blockchain

the current block is discarded. Generally speaking, in the Bitcoin system, the transactions on the six blocks before the current block are basically considered to be accepted by the entire system in terms of probability.

The biggest feature, as well as the advantage of PoW, is reflected in the fairness of the protocol, which is if a miner's computing power accounts for $p\%$ of the network's total computing power, there is a corresponding $p\%$ possibility to generate blocks and get paid. That also illustrates the difficulty of an attack. The attacker's computing power needs to compete with other honest nodes in the whole network to generate the blocks that are "beneficial" for him. The PoW algorithm has successfully guaranteed the safety of the Bitcoin network from birth.

However, as more and more people use Bitcoin for trading, its defects are gradually manifested. The original intention of PoW is to achieve a decentralized democratic consensus through "one-CPU-one-vote", which is a time-consuming process. In addition, due to the fast increase in Bitcoin prices, many types of professional mining equipment appear on the market. The increase in the number of users purchasing mining equipment leads to the loss of more and more ordinary miners. The foundation of democracy is damaged, and monopoly issues are also highlighted [34]. As more and more users participate in Bitcoin mining, in order to reduce the mining threshold, but also to improve the stability of mining, many commercial mining pools occur in the system. A mining pool is an open mining server which forces many users' computing power to a team to mine, such as BTC.COM, AntPool, SlushPool, etc.[1]

As shown in Fig. 3, over the past 24 hours of October 14th, 2019, nearly 50% of the blocks were mined by the top three mining pools. It is undeniable that mining pools have mastered enormous computing power. If a single mining pool exceeds 50%, or several large mining pools make an alliance privately, it is easy to launch a 51% attack on the Bitcoin system.

Secondly, the problem of energy waste has been criticized for a long time. Numerous mining rigs waste a lot of electric power day and night, but have no other effect except generating Bitcoins. Table 1[2] indicates the energy consumption statistics of the Bitcoin network currently. It is estimated that the Bitcoin system has consumed at least 73.12 TWh of electricity annually, making it comparable with the amount of a country such as Austria [20].

Besides, the PoW consensus mechanism has some other problems such as long confirmation cycle, and low throughput. Regarding the problems of the Nakamoto Consensus, blockchain systems have conducted different improvements based on specific conditions. There are two ways of improvement. One is the engineering improvement, e.g., the improvement of Primecoin[3] is an algorithm to turn meaningless hashing into a meaningful search for large prime numbers when seeking *Nonce*. It is expected to bring some scientific contributions to mathematical academia. Focusing on the increasing centralization of computing power caused by ASIC (Application Specific Integrated Circuit) mining rigs, Tromp [50] proposed an anti-ASIC mining rig algorithm based on memory consumption. The other venue of improvement is to change the consensus mechanism, such as the Proof of Stake (PoS) and Delegated Proof of Stake (DPoS), which are mostly adopted and will be discussed below.

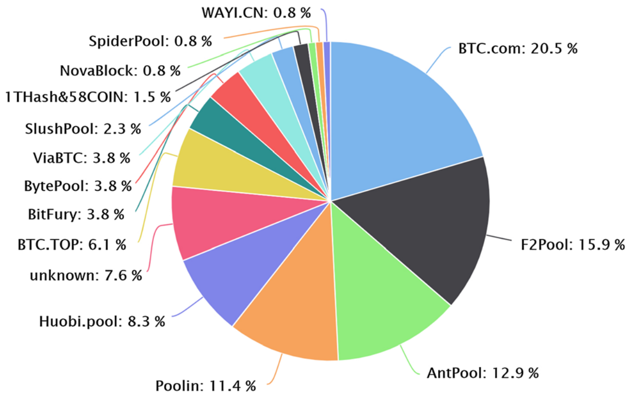


Fig. 3 Computing power distribution of current mining pools

2.2 PoS

Owing to the vulnerabilities like the serious waste of computing power and the 51% attack in the PoW mechanism, researchers have put forward a new kind of consensus mechanism known as Proof of Stake (PoS) [29]. What is the "stake"? In early versions of PoS, it has another commonly-used name "coin age", i.e., currency amount times holding period. For example, if Alice received two coins from Bob and held it for 50 days, then Alice has accumulated 100 coin-days of coin age (2×50). And when Alice spent the coins, we say the collected coin age had been consumed. Nodes with a positive stake are called stakeholders. In contrast to PoW's ability to compete for recording data in accordance with the ability of each node, PoS has more ability to record data for those nodes with more stakes (or coin age). The manifestation of this ability is that for a node with a longer coin age, its bookkeeping difficulty is relatively lower.

[1] Global computing power distribution [Online], available: https://btc.com/stats/pool, October 14, 2019.

[2] Energy consumption statistics [Online], available: https://digiconomist.net/bitcoin-energy-consumption, October 14, 2019.

[3] Primecoin Website [Online], available: http:// primecoin.io/, October 14, 2019.

Table 1 The energy consumption statistics of the Bitcoin network

Description	Value
Bitcoin's current estimated annual electricity consumption (TWh)	73.12
Bitcoin's current minimum annual electricity consumption (TWh)	52.1
Annualized global mining revenues	$5,839,927,943
Annualized estimated global mining costs	$3,656,073,069
Current cost percentage	62.60%
Estimated electricity used over the previous day (KWh)	200,332,771
Implied Watts per GH/s	0.086
Total Network Hashrate in PH/s (1,000,000 GH/s)	97,145
Electricity consumed per transaction (KWh)	624
Number of U.S. households that could be powered by Bitcoin	6,770,506
Number of U.S. households powered for 1 day by the electricity consumed for a single transaction	21.08
Bitcoin's electricity consumption as a percentage of the world's electricity consumption	0.33%

In order to generate blocks faster, the PoS mechanism replaces the process of exhaustively seeking *Nonce* with the algorithm below:

$$H(H(B_{prev}, A, t)) \leq balance(A)m \tag{2}$$

Here, H is still a hash function, t is the UTC timestamp, B_{prev} refers to the previous block, $balance(A)$ is the coin age of the account A and m is a fixed real number.

Peercoin (PPC) [29] is the first to introduce the PoS mechanism into the blockchain system in 2012. In PPC, in addition to processing classical PoW-based transactions, the system also deals with a kind of transactions called coin-stake in which each transaction will consume the coin age of data record. In coin-stake transactions, each stakeholder is required to send coins to himself (to ensure that the coin age clears to zero after the stake block is generated), which is used to generate a PPC block and obtain partial revenue. The cost of gaining revenue is the consumption of coin age. Similar to the Bitcoin system, the PPC block also requires participants to look for random numbers to make the hash value of block header meet the target difficulty, except that the target difficulty to generate a block in PPC system is different for various participants. The target difficulty is inversely proportional to the coin age consumed in coin-stake. The more coin age accumulated by participants, the lower the bookkeeping difficulty, and the greater the probability of generating blocks. In other words, the concept of coin age in PoS can be imagined as the computing power in PoW. If someone holds a large sum of currency for a long time, then he will have the opportunity to use a powerful ASIC mining rig once in the next mining process. But this opportunity does not depend on the consumption of hardware and electricity, it only depends on the user's deposit in the system and the time of saving the currency. Unlike the competition in PoW mining, PoS mining is more like a lottery. The more accumulated the coin age, the more chance there is to win. Once the winning is already, the coin age will be consumed, and the probability of a second win will be reduced [26].

The transformation of the design basis brings PoS the following advantages.

Firstly, PoS alleviates the waste problem of PoW mining. In the Bitcoin system, the probability of generating blocks is directly proportional to the miners' workload. In a PoS system, the probability of block generation is proportional to the coin age. Therefore, miners no longer need to invest heavy computing power to win the competition.

Secondly, it is more difficult for the adversary to attack the cryptocurrency system. In PoS, the main chain is defined as the chain that consumes the most coin age. Each block's transaction will submit the consumed coin age to this block to increase the probability. In this case, if the adversary wants to initiate an attack on the main chain, he must own a large sum of coins, and accumulate enough coin age. The cost of getting a large sum of coins in the PoS system is higher than the cost of mastering most of the computing power in the PoW system. Besides, once the attack is implemented, not only the system will be destroyed, but also the wealth the attacker owns will be damaged. This may reduce the attacker's motives from the beginning. And once the block is generated, the coin age will be immediately cleared, which also guarantees that the attacker cannot continue the attack [7].

However, the PoS consensus mechanism is not perfect as well.

The first is the distribution of the initial currency. Currently, the cryptocurrency systems using PoS have two methods to supply the initial currency. One is to use PoW for the early stage of mining and then use PoS for system maintenance. The other is IPO (Initial Public Offerings), but lack of trust. The currency is concentrated in the hands of developers and a few people, unlike everyone in the PoW mechanism has the opportunity to get coins.

Second, PoS encourages the behavior of hoarding. The coin-stake transaction in PoS generates blocks and benefits by destroying the coin age, but the coin age of other common transactions packaged into the block is also reset to zero. This coin age does not bring stakeholders the benefit. It just disappears in vain for them.

The third is since the coin age will also accumulate when the node is offline, the node may prefer not to go online until the coin age has accumulated to a certain extent [32]. Lack of enough online nodes will make it easy to launch network attacks. Besides, due to the lack of online nodes, the speed of data synchronization and transaction response will be affected.

The next problem is the costless simulation. This suggests that in the absence of PoW, PoS is proof of a virtual resource. There is nothing that prevents users from doing it over and over, perhaps in parallel multiple times. In PoW, all the parties must commit to the execution of consensus and advance that execution. This is not the case in PoS, because it is “nearly” costless to execute PoS protocol. In principle, there is virtually nothing at stake and one would be capable of advancing multiple different executions of the protocol so that it can find the more favorable one. That could be lead to the so-called “nothing-at-stake” attack. Take a look at Fig. 2 for more illustration. If one is a validator, then he can simply put his money in both the blue chain and green chain without any fear of repercussion at all. No matter what happens, he will always win and have nothing to lose, despite how malicious his actions may be.

The last is the “long-range” attack. In long-range attacks, there is a victim node that tries to distinguish between two alternative histories without access to recent information. If a node is constantly online, it is easy to know about what happens in the network. But if the node joins the network after a big hiatus or it is a new node, then the bootstrapping problem may arise. It is difficult for it to synchronize with the blockchain without any recent information.

2.3 DPoS

In order to further speed up the transaction and solve the security problem that the offline node in the PoS can also accumulate the coin age, Daniel Larimer proposed DPoS (Delegated Proof of Stake) in April 2014 [49], which is currently the consensus mechanism for BitShares [33] and Crypti [15] platforms. In DPoS, the system introduces two roles called witness and delegate, both of which have multiple members. The candidates of these two roles are selected by the stakeholders with an approval voting process according to the number of their stakes. Stakeholders with more than 51% stakes can vote for the N witnesses and delegates. The witnesses themselves are irrelevant to the transaction accounts they participate in. They only participate in the block generation and obtain revenue from transaction fees. As the joint signers of the stakeholder’s account, delegates are responsible for adjusting the parameters such as the process of generating the block of the witness and the transaction fees. The adjustment is performed under the supervision of the stakeholders. Compared with the node feature of PoS that each node has equal rights to generate a block, nodes of DPoS are divided into delegates and witnesses, which have different rights respectively. As shown in Fig. 4, the delegates are responsible for voting and the witnesses just need to be their follower nodes. That is the critical difference between PoS and DPoS.

DPoS mechanism is similar to the decision of the board of directors in the real world. Stakeholders vote for a delegate. The system calculates a certain number of delegates with the most votes based on the stakes of stakeholders, and the delegate takes turns to generate the block in a prescribed order. After voting by all stakeholders, the trust in the system has been concentrated by a small number of participants, and the node does not have to wait for confirmation of a considerable number of untrusted nodes after the transaction is initiated, but only the delegate needs to verify the transaction. This voting mechanism concentrates the power of all users in the hands of a few people, but greatly shortens the confirmation time of transactions. Compared with the PoW-based system, the block generation time is shorter, and the throughput has been greatly improved. Taking BitShares as an example, its peak throughput can be thousands of transactions per second. The confirmation time is reduced to the seconds, which brings cryptocurrency technology to a new level.

In another version of DPoS protocol, the node has to pay a price to become a delegate, such as paying a deposit to a security account. If the node does something evil, the

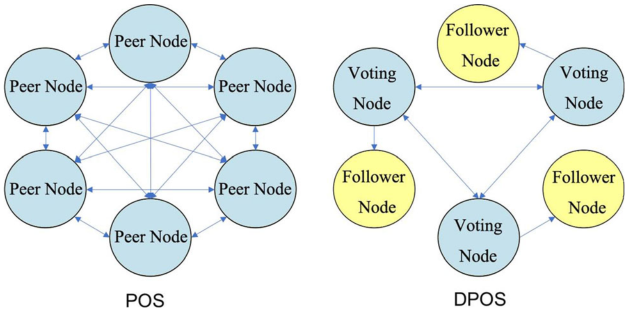


Fig. 4 Node differences between PoS and DPoS

deposit will be confiscated [33]. Conversely, if the delegate maintains the system well, he will share the block transaction fee with other delegates, so that the reward will positively encourage the delegate to work harder to maintain system security. Since the block is signed by the delegates in turn, if a delegate is offline and misses signing the block, he will face the risk of being replaced by other candidate delegates. Therefore, the delegate must guarantee sufficient online time for the profit. This version of DPoS protocol is also known as a deposit-based proof of stake.

2.4 PBFT

The aforementioned protocols all belong to the permissionless consensus protocols, which means that the number of distributed nodes involved cannot be predicted. When multiple participants of a distributed system intend to modify the state of the system through additional blocks, they cannot simply determine it via the mechanism that most people make decisions. These update operations can only be optimized by PoW or PoS. For those scenarios in which the participants are relatively fixed, nodes of the distributed system have been determined in advance. Therefore, the majority rule can be selected. PBFT (Practical Byzantine Fault Tolerance) [22] is a permissioned protocol that participants determine and agree on the majority rule. It was proposed by Miguel Castro and Barbara Liskov in 1999. Before introducing that, we need to know the original BFT mechanism.

Nodes with Byzantine failures are called Byzantine nodes, while other nodes are non-Byzantine nodes. The BFT system satisfies the following conditions for each request: all non-Byzantine nodes use the same input information to produce the same result; if the input information is correct, then all non-Byzantine nodes must receive this information and calculate the corresponding result.

The assumptions commonly used by the Byzantine system include:

(1) The behavior of the Byzantine nodes can be arbitrary, and the Byzantine nodes can collude.
(2) Errors between nodes are irrelevant.
(3) Nodes are connected through an asynchronous network, and the messages in the network may be lost, out of order or delayed, but most protocols assume that the message can be delivered to the destination in a limited time.
(4) The message transmitted between the servers can be sniffed by the third party, but the third party can not falsify the content of it or verify the integrity of it.

The original BFT system lacks practicality due to the need to demonstrate its theoretical feasibility. Also, an additional clock synchronization mechanism is required, and the complexity of the algorithm increases exponentially as nodes increase.

Compared to the traditional BFT algorithm, PBFT reduces the time complexity from exponential to polynomial, which not only greatly improves efficiency, but also makes it the first widely-used Byzantine consensus algorithm. It can resist a certain number of Byzantine nodes in the system. In a PBFT-based blockchain system, the system that tolerates f Byzantine fault nodes needs at least $3f + 1$ participating nodes and then reaches a consensus in polynomial time. From the practical perspective, PBFT is now the default consensus algorithm of a famous blockchain project, Hyperledger, hosted by the Linux Foundation [9].

The PBFT consensus divides nodes into two types: primary nodes, which are responsible for sorting the client's requests, and the rest are backup nodes, which execute the requests in the order provided by the primary node. The algorithm specifies three basic protocols: agreement, checkpoint, and view change. The agreement is to ensure that requests from clients are executed in a fixed order on each server. It contains five stages: request, pre-prepare, prepare, commit and reply. Usually, a consensus process will be performed in the same view. However, when the primary node fails, the view-changing protocol replaces the primary node with the backup node in sequence and ensures that the request that has been executed by the normal node is not tampered with. During the consensus process, the node records the log at any time. If the log is not cleaned up in time, the system resources will be occupied by useless information, which will affect the overall performance. At the same time, the states of different nodes may be inconsistent because the asynchronous nature of the system cannot guarantee that each node performs the same request. Therefore, the checkpoint protocol is executed periodically to handle the log and correct node status.

The PBFT consensus is generally suitable for private blockchain and consortium blockchain scenarios where the source of nodes is relatively reliable. It has many advantages:

- The operations of PBFT-based system can be separated from the existence of currency. The consensus nodes are composed of business participants or supervisors, hence the security and stability are guaranteed by the business-related parties. But the PoW, PoS, and DPoS system cannot be separated from the existence of currency. Their systems must have a reward mechanism for the currency and the security of systems is guaranteed by the holders of the system currency.

However, when a blockchain system is actually applied in commerce, the value of the assets carried by it may far exceed the value of the currency issued by it and it will be unreliable to let stakeholders guarantee the security and stability of it.

- The delay of the PBFT consensus protocol is about 2 s to 5 s, which basically meets the requirements of commercial real-time processing scenarios.

As for the weaknesses, PBFT is a weakly synchronous protocol, so it relies critically on network timing assumptions, and only guarantees liveness when the network behaves as expected.

To improve that, Andrew Miller proposed the HoneyBadgerBFT [38], the first practical asynchronous BFT protocol which guarantees liveness without making any timing assumptions, in 2016. The core process of HoneyBadgerBFT consists of "Atomic Broadcast" and "Asynchronous Common Subset". It uses N binary consensus protocol instances and determines a common subset based on the instance results. For higher efficiency, HoneyBadgerBFT adopts two methods: (1) mitigate single-node bandwidth bottleneck by splitting transactions; (2) improve transaction throughput by selecting random trading blocks in batch transactions and matching threshold encryption. Experiments [38] show that its efficiency is significantly increased compared with the traditional PBFT consensus.

2.5 Other consensus protocols

The four mentioned above are the common consensus protocols adopted by the current blockchain systems and all have actual implementations as support. However, the analysis shows that there are some potential flaws in these incipient consensus protocols. In recent years, many researchers have conducted in-depth research on the consensus problem and proposed some new algorithms. Among them, we introduce several representative algorithms with better performance, including Ripple [45], Proof of Activity (PoA) [6], Algorand [25], Snow White [17], Casper [8] and Ouroboros Genesis consensus [4].

2.5.1 Ripple

Ripple is an Internet-based open-source payment protocol that enables decentralized currency exchange, payment and clearing functions. In Ripple's network, transactions are initiated by the client (application) and broadcasted to the entire network via tracking nodes or validating nodes. The main function of the tracking node is to distribute transaction information and respond to the client's ledger request. The validating node can add new data to the ledger through the consensus protocol.

Ripple's consensus is achieved between the validating nodes. Each validating node is pre-configured with a list of trusted nodes called UNL (Unique Node List). Nodes on the list can vote on the transaction. In Ripple's consensus algorithm, the identity of nodes participating in the voting has been known in advance. Therefore, it is more efficient than many anonymous consensus algorithms such as PoW, with a few seconds to confirm the transaction. Of course, Ripple is only suitable for the permissioned chain. The BFT capability of it is $(n-1)/5$, which can tolerate the Byzantine faults of 20% nodes in the entire network without affecting the correct consensus.

2.5.2 PoA

The Proof of Activity (PoA), proposed by Bentov et al, combines the characteristics of PoW and PoS. PoW could lead to the centralization of computing power, while PoS/DPoS tends to form an oligarchy of stakes due to the scale effect of stakes. The centralization of computing power or stakes poses a potential threat to the safety and stability of the blockchain systems.

Miners in the PoW system are pursuing the maximization of their interests. For higher economic benefits, the security of the cryptocurrency network may be jeopardized, and the stakeholders are suitable to help accomplish this task. Based on this assumption, the basic idea of the PoA's ability to prevent excessive centralization of computing power and stakes is to allow participants in the transaction to participate more in the generation of blocks to counterbalance the dominant miners.

The implementation of PoA is as follows. The miner generates a new block header that satisfies the difficulty, and the header includes the hash value of its predecessor and the information of N traders involved in the possible new block. After mining the block header, miners broadcast the (possible) new block header. Relevant stakeholders and participants of N transactions use their private keys to sign the transactions, and the last-signed trader packs the block, then broadcasts it and participates in the bookkeeping competition as traditional Bitcoin does. Through this process, miners and trading participants share the revenue of ledger. The signature of these N participants is the Proof of Activity. The advantage is that miners who dominate the computing power are not able to monopolize the bookkeeping ability without the cooperation of traders (as it cannot be signed by their private keys).

2.5.3 Casper

Casper is a security-deposit based PoS protocol prepared for Ethereum v2.0 [58], a blockchain-based distributed computing platform and operating system. To address the nothing-at-stake attack of PoS, Casper has implemented a process; in this way, they can pass away all malicious elements. This is how PoS works under Casper: The validators take some parts of their Ethers (a.k.a. tokens issued by Ethereum) as stakes. After that, they begin to validate the blocks, i.e., when they discover a block which can be regarded to be added to the chain, they will validate it by placing a bet on it. If the block is appended, then the validators will get a reward proportional to their stakes. However, if a validator performs maliciously and tries to perform a "nothing at stake", he will immediately be dressed down, and all of his stakes will be slashed.

Casper is designed to work in a trustless system and be more Byzantine Fault Tolerant. Anyone who performs maliciously will be immediately punished with his stakes being slashed off. This is the most unique feature it differs from other PoS protocols. Moreover, Casper has more critical incentives to ensure network security, including punishing miners who perform offline, involuntarily or not. This indicates that validators have to be careful about node uptime. Carelessness or laziness will result in the loss of their stakes. This property alleviates the censorship of transactions and the entire availability.

2.5.4 Snow white

Snow White is a PoS derivative consensus protocol adopting the ideas of a simpler protocol dubbed Sleepy [42]. Sleepy aims to achieve the guarantees of chain growth, chain quality, and consistency with 51% of online nodes. It is designed for deployment in a permissioned context and relies on the assumption on stake assigned or instantiated by some trusted sources. Every second, every member of the committee is eligible to mine a new block in the system, which involves a standard block mining solution with a public source of entropy as the *Nonce*. The challenges of choosing a suitable mining function and source of entropy are addressed in the work, and the proof is given that no committee member can manipulate the protocol to get profit.

Snow White, on the other hand, is an extension of Sleepy intended to provide the same rigorous blockchain-derived guarantees in a permissionless setting. The problem is apparently much more difficult: it is nontrivial to choose suitable committee members for the block lottery and ensure no coalition of the committee members to get profit. The solving protocol is simple: in each step, a committee mines as in Sleepy, with a shared source of entropy h_0. With enough bits of entropy in h_0 and an appropriately selected committee weighted on stake, it is possible to prove the desired result of chain quality, growth, and consistency. Choosing both the committee and h_0 such that no adversary gain substantial advantages by deviating from the protocol is the key to the construction and concrete parameters of the protocol.

In Snow White, it assumes an optimal adversary with the ability to delay network messages up to some arbitrary time, and a very strong notion of an attacker that makes it the most rigorously conceived protocols thus far, in both the permissioned and permissionless PoS scenarios. As for the performance, Snow White achieves comparable transaction confirmation time and throughput as PoW blockchains while completely dispensing with the wasteful computation during the simulation experiments.

2.5.5 Algorand

The Algorand consensus is a new consensus based on PoS and cryptology. The name "Algorand" is synthesized by two words: algorithm and random, meaning that it is a public ledger protocol based on a random algorithm. According to its analysis, Algorand has the characteristics of short agreement time, strong anti-attack ability, low computing power, and better economic profits.

Algorand employs a similar concept of "Write-Ahead Logging" in the traditional database. In Algorand, the consensus towards a new block is reached through a Byzantine agreement called $BA*$. Generally speaking, the execution of $BA*$ consists of two phases: (1) synchronously determine the highest-priority block; (2) reach consensus on two options: either to agree on a proposed block or agree on an empty block. Each phase has several steps. The process for the first phase is shown in Fig. 5. Algorand can reach consensus within roughly one minute.

Algorand divides the participants into two roles: leaders and verifiers. Both roles are uncertain and based on the previous block. That is, before each block is generated, a batch of potential leaders is generated first. These leaders

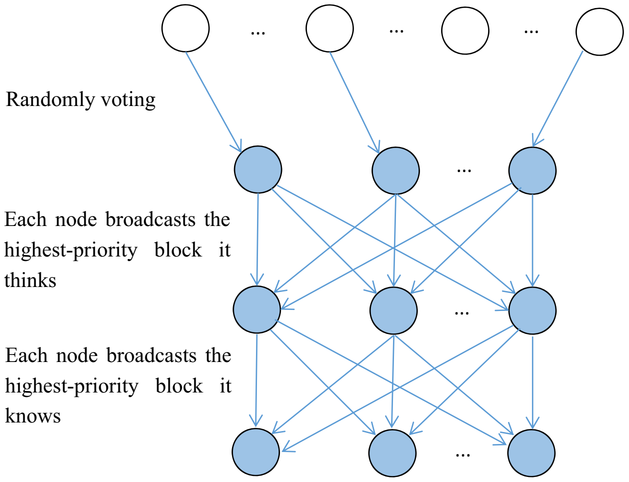

...
...
...
Randomly voting
...
Each node broadcasts the highest-priority block it thinks
...
Each node broadcasts the highest-priority block it knows
...

Fig. 5 The first phase of $BA*$

know and can prove to the entire system that they are the producers of a candidate block. Each potential leader generates a candidate block and attaches its one-time signature and signature public key to the entire system for verification. At last, the verifiers vote for the determination of whether the block generated by the leader will be adopted or not. Once the verifiers have reached a consensus on a new block, more than half of the verifiers will sign the block with their own private keys, and the block will be broadcasted in the Algorand network.

2.5.6 Ouroboros genesis

Ouroboros Genesis is a PoS-based consensus protocol that provides security against fully-adaptive corruption in the semi-synchronous setting for the first time. It is the third version of the Ouroboros consensus, and the first provable secure and robust PoS algorithm proposed in 2017. The former Ouroboros protocols creatively design secure random numbers to elect unpredictable block creators [18, 28]. This randomness allows an unbiased slot leader (block creator) selection process to select a leader with a probability proportional to its stake. During an epoch (a regular interval divided into many time slots), stakeholders execute the coin-flipping protocol and finally select the slot leader for the next epoch. Compared with the former versions, the biggest improvement of Ouroboros Genesis is to solve the problem of long-range attack aroused by original PoS consensus.

In a long-range attack, a new or offline node without any information is trying to find "which is the right history". Suppose most honest nodes provide the real blockchain, and an attacker provides another. But the new or offline node only holds the genesis block. However, Ouroboros Genesis proposes a novel chain selection rule, showing that the problem of an attacker reusing an opportunity to issue a block in multiple paths of a fork can be overcome. New or offline nodes can securely join the right chain and enable their blockchains from the genesis block [4]. Researchers prove that adversarial blockchains shortly after the divergence point will isolate a certain region of blocks and exhibit a less dense block distribution. Within a certain time range, the node is going to follow the chain that is denser. The rule in this consensus is quite simple to implement by program and it will enhance the longest chain rule.

3 Consensus comparison

Through the aforementioned introduction to blockchain consensus protocols, it is evident that each protocol has a different emphasis on design, thus presenting diverse advantages and disadvantages. For a better comparison of these particularities, we briefly introduce four major dimensions to evaluate them qualitatively and quantitatively and discuss their performances in Table 2 and figures. In view of the purpose and effect of consensus, we can evaluate from the following aspects:

- Security. It means the capability of fault tolerance which is mostly based on the design principle of a consensus mechanism. Here, Fig. 6 shows the tolerated power (of computing power, validators, stake or other adversaries) of these consensus protocols. On the other side, it can be evaluated from two important properties in the distributed system, consistency (safety) and liveness [30]. Consistency means nodes can eventually reach the same local state. Liveness means transactions will always be processed in a limited time. Ensuring consistency and liveness in an asynchronous network at the same time is difficult, so a consensus designer usually chooses to guarantee one and give up another in specific situations, such as the preference of PoW is liveness, and BFT-based protocols prefer consistency.
- Scalability. The ability to support the expansion of node numbers. Network scalability is one of the key factors to be considered in the design of a blockchain. It can be generally observed from throughput, which defines the number of transactions the system can process per second. Most consensus protocols have poor scalability. For example, the Bitcoin platform supported by PoW handles up to 7 transactions per second, which is far from the performance of the existing centralized trading systems.
- Performance. Throughput and latency. Figure 7 reveals the peak throughput of each consensus. Latency metrics include block time and translation latency. Block time is defined as the time it takes to mine a block. Transaction latency is the time required for a transaction from initiation to confirmation by consensus in the system, and it contains block time. There are many factors affecting throughput and latency, such as the number of consensus nodes, the complexity of the message, the time required for message validation, the bandwidth available for consensus, etc. In Fig. 8, the average block time and transaction latency of each consensus is shown.
- Energy consumption. We are also concerned about the energy consumed by each node for reaching the consistency of a transaction under the guidance of consensus, including CPU, memory, battery, etc.

Since PBFT and Ripple require permission, there is a limit to the number of participating nodes. Also, PBFT's scalability is relatively low comparing to other consensus protocols. In terms of energy consumption, due to the need

Table 2 Blockchain consensus comparisons

Consensus	PoW	PoS	DPoS	PBFT	Ripple
Number of nodes	Unlimited	Unlimited	Unlimited	Limited	Limited
Permission	No	No	No	Yes	Yes
Scalability	High	High	High	Low	High
Energy consumption	High	Low	Low	Low	Low
Safety preference	Liveness	Liveness	Liveness	Consistency	Consistency
Latency	High	High	Low	Low	Low
Throughput	Low	Low	High	High	High
Example	Bitcoin	Peercoin	BitShares	Hyperledger Fabric v0.6	Ripple

Consensus	PoA	Casper	Snow White	Algorand	Ouroboros Genesis
Number of nodes	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Permission	No	No	No	No	No
Scalability	High	High	High	High	High
Energy consumption	Low	Low	Low	Low	Low
Safety preference	Consistency	Consistency	Liveness	Consistency	Liveness
Latency	Low	High	High	Low	Low
Throughput	Low	High	Low	High	Low
Example	Decred	Ethereum v2.0	None	ArcBlock	Cardano

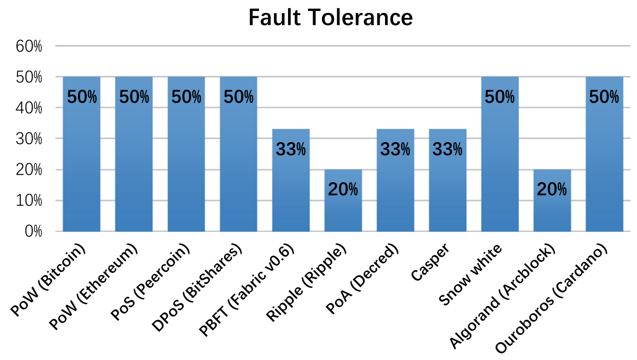


Fig. 6 Fault tolerance (percentage) of consensus protocols

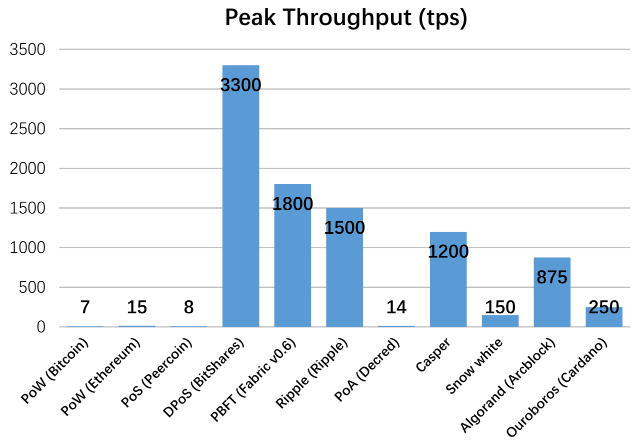


Fig. 7 Peak throughput (transaction per second) of consensus protocols

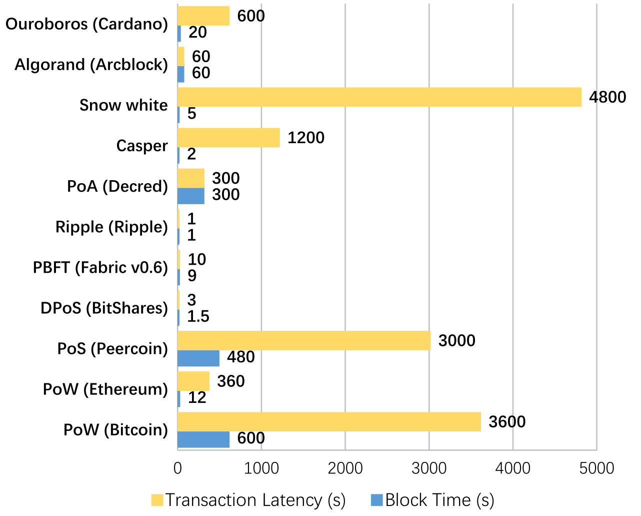


Fig. 8 Average block time and transaction latency (per second) of consensus protocols

for complex hash computing, PoW thus has the largest energy consumption and the energy consumptions of other consensus protocols are relatively lower. Due to the intervals of blocks and the need for multiple confirmations, the throughput of PoW and Snow White is far lower than other consensus protocols.

Each consensus has its own shortcomings. As for the systems based on PoW, the more incentives the nodes with stronger computing power gain, the more centralized the computing power tends to be. PoS and DPoS systems also have similar problems with the centralization of stakes. New consensus protocols like Algorand can theoretically avoid the above situations, but taking time to verify the practical effects of them. Although the blockchain system based on PBFT does not have the problems of computing power and stake centralization, its scalability is limited and the delay and throughput will decrease significantly as the number of nodes increases.

4 Future directions

In this section, we present some research trends in consensus studies.

Diversification of proof methods Early PoW and PoS mechanisms have the problems of waste of resources and low initiative of nodes. Researchers have developed Proof of Time [13], Proof of Storage [37], Proof of Existence [14], Proof of Contribution [61], Proof of Authority [19], Thunderella [43], Proof of Play [64], Proof of Elapsed Time [12], Proof of Luck [39], Proof of DDoS [59], Proof of Sincerity [65] and other mechanisms for the purpose of reducing the cost of mining competition or improving resource utilization and application scenarios. The new proof methods will continue to emerge. However, when designing consensus algorithms, the key points are to make the mining power sufficiently dispersed, to increase the difficulty of attackers to master most of the competitiveness, and to reduce the possibility of individual nodes or organizations rewriting the blockchain. In this way, we can effectively prevent the double spend attack and ensure the security of the system.

Hybridization of different consensus mechanisms The threat of PoW comes from miners with high computing power, and the security risks of PoS are active major stakeholders. Researchers suggest to combine PoW with PoS, so if someone wants to launch a 51% attack, the malicious node needs to master most of the computing power and most of the stakes, which becomes a more difficult condition to achieve. If someone does this, the entire blockchain system will be destroyed due to excessive centralization. Based on this idea, we can design a protocol which must provide a proof of work that meets certain difficulty to participate in the consensus, select the block creator through the verifiable random function [25], and reach a consensus through BFT thoughts.

Designing reasonable incentives In blockchain systems, incentives are often introduced to deal with technical problems. For example, the IPFS [5] technology for solving the blockchain storage problems is also a combination of incentive mechanisms to encourage users to assist in storing data fragments before they can form a complete project Filecoin. Therefore, if we combine the specific processes of consensus and design more reasonable incentive measures, we will achieve twice the result with half the effort in actual operation, and will also have a positive effect on the safety and continuity of the system. In addition, researchers have been arguing whether there is a need for internal tokens in the consortium chain. Some researchers argue that it is necessary to add coins to implement reward and punishment functions in some consortium chains with incomplete trust. Through the continuous exploration of more researchers, we believe that there will be more solutions suitable for the consortium chain incentives in the future.

5 Conclusion

In recent years, as the blockchain technology has received extensive attention, consensus algorithms have been studied by more and more researchers. As the most important part of the blockchain, the consensus algorithm embodies the performance and functionality of the blockchain system. In this paper, we present a classification of current consensus protocols in the blockchain system, enumerate the particularities of mainstream protocols (PoW, PoS, DPoS, PBFT, etc.) and analyze the pros and cons of them. Then we compare the performance of them qualitatively and quantitatively. Our evaluation result shows that the performance of current consensus protocols is still far from the industrial need. Therefore, researchers should develop more practical consensus protocols.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61872416, 51879210, 51479159, 61671216, 61871436, 61872415 and 61702204; by the Fundamental Research Funds for the Central Universities of China under Grant 2019kfyXJJS017; and by the fund of Hubei Key Laboratory of Transportation Internet of Things under Grant 2018IOT004.

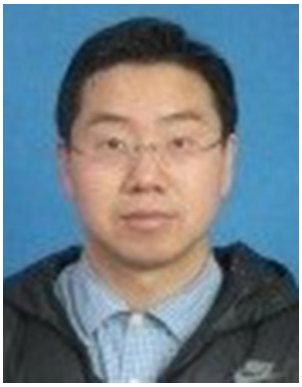
References

1. Zhang, R., Xie, P., Wang, C., Liu, G., & Wan, S. (2019). Classifying transportation mode and speed from trajectory data via deep multi-scale learning. *Computer Networks*, *162*, 106861.
2. Adhikari, A., Rawat, D.B., & Song, M. (2019). Wireless network virtualization by leveraging blockchain technology and machine learning. In: *Proceedings of ACM workshop on wireless security and machine learning* (pp. 61–66).
3. Back, A. (2002). Hashcash - a denial of service counter-measure. In: *Proceedings of USENIX annual technical conference*.
4. Badertscher, C., Gaži, P., Kiayias, A., Russell, A., & Zikas, V. (2018). Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In: *Proceedings of ACM SIGSAC conference on computer and communications security* (pp. 913–930).
5. Benet, J. (2014). IPFS - content addressed, versioned, p2p file system. Eprint Arxiv.

6. Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake. *ACM SIGMETRICS Performance Evaluation Review*, *42*(3), 34–37.
7. BitFury, G. (2015). Proof of stake versus proof of work white paper. Retrieved September 27, 2019 from http://bitfury.com/content/5-white-papers-research/posvs-pow-1.0.2.pdf.
8. Buterin, V., & Griffith, V. (2017). Casper the friendly finality gadget. arXiv preprint arXiv:1710.09437.
9. Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In: *Proceedings of workshop on distributed cryptocurrencies and consensus ledgers* (Vol. 310, p. 4).
10. Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. arXiv preprint arXiv:1707.01873.
11. Chen, K., Wang, C., Yin, Z., Jiang, H., & Tan, G. (2018). Slide: Towards fast and accurate mobile fingerprinting for wifi indoor positioning systems. *IEEE Sensors Journal*, *18*(3), 1213–1223.
12. Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., & Shi, W. (2017). On security analysis of proof-of-elapsed-time (poet). In: *Proceedings of international symposium on stabilization, safety, and security of distributed systems* (pp. 282–297).
13. CHRONOLOGIC: Chrono logic whitepaper (2017). Retrieved September 27, 2019 from https://chronologic.network/uploads/ChronologicWhitepaper.pdf.
14. Crosby, M., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. *Applied Innovation*, *2*(6–10), 71.
15. Crypti: Crypti whitepaper (2015). https://bravenewcoin.com/insights/crypti-white-paper .
16. Dai, Y., Xu, D., Maharjan, S., Chen, Z., He, Q., & Zhang, Y. (2019). Blockchain and deep reinforcement learning empowered intelligent 5g beyond. *IEEE Network*, *33*(3), 10–17.
17. Daian, P., Pass, R., & Shi, E. (2016) Snow white: Provably secure proofs of stake. Cryptology ePrint Archive, Report 2016/919. https://eprint.iacr.org/2016/919.
18. David, B., Gaži, P., Kiayias, A., & Russell, A. (2018). Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In: *Proceedings of annual international conference on the theory and applications of cryptographic techniques* (pp. 66–98).
19. De Angelis, S., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., & Sassone, V. (2018). Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain.
20. De Vries, A. (2018). Bitcoin's growing energy problem. *Joule*, *2*(5), 801–805.
21. Dwork, C., & Naor, M. (1993). Pricing via processing or combatting junk mail. In: *Proceedings of international cryptology conference on advances in cryptology* (pp. 139–147).
22. Fan, J., Yi, L. T., & Shu, J. W. (2013). Research on the technologies of Byzantine system. *Journal of Software*, *24*(6), 1346–1360.
23. Gao, H., Huang, W., Yang, X., Duan, Y., & Yin, Y. (2018). Toward service selection for workflow reconfiguration: An interface-based computing solution. *Future Generation Computer Systems*, *87*, 298–311.
24. Gao, H., Mao, S., Huang, W., & Yang, X. (2018). Applying probabilistic model checking to financial production risk evaluation and control: A case study of alibaba's yu'e bao. *IEEE Transactions on Computational Social Systems*, *5*(3), 785–795.
25. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling Byzantine agreements for cryptocurrencies. In: *Proceedings of 26th symposium on operating systems principles* (pp. 51–68).
26. Houy, N. (2014). It will cost you nothing to 'kill' a proof-of-stake crypto-currency (Vol. 34, no. 2). New York: Social Science Electronic Publishing.
27. Huang, H., Yin, H., Min, G., Zhang, J., Wu, Y., & Zhang, X. (2018). Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks. *IEEE Transactions on Mobile Computing*, *17*(6), 1339–1352.
28. Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In: *Proceedings of annual international cryptology conference* (pp. 357–388).
29. King, S., & Nadal, S. (2012). PPCoin: Peer-to-peer crypto-currency with proof-of-stake. Technical Report. https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf
30. Lamport, L. (1977). Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, *2*, 125–143.
31. Lamport, L. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages & Systems*, *4*(3), 382–401.
32. Lampson, B.W. (1996). How to build a highly available system using consensus. In: *Proceedings of international workshop on distributed algorithms* (pp. 1–17).
33. Larimer, D., & Kasper L, S.F. (2015) Bitshares 2.0: Financial smart contract platform.
34. Laszka, A., Johnson, B., & Grossklags, J. (2015). When bitcoin mining pools run dry. In: *Proceedings of international conference on financial cryptography and data security* (pp. 63–77).
35. Liu, Y., Hao, L., Liu, Z., Sharif, K., Wang, Y., & Das, S. K. (2019). Mitigating interference via power control for two-tier femtocell networks: A hierarchical game approach. *IEEE Transactions on Vehicular Technology*, *68*(7), 7194–7198.
36. Liu, Y., Quan, W., Wang, T., & Wang, Y. (2018). Delay-constrained utility maximization for video ads push in mobile opportunistic d2d networks. *IEEE Internet of Things Journal*, *5*(5), 4088–4099.
37. Miller, A., Juels, A., Shi, E., Parno, B., & Katz, J. (2014). Permacoin: Repurposing bitcoin work for data preservation. In: *Proceedings of IEEE symposium on security and privacy*.
38. Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The honey badger of bft protocols. Cryptology ePrint Archive, Report 2016/199. https://eprint.iacr.org/2016/199.
39. Milutinovic, M., He, W., Wu, H., & Kanwal, M. (2016). Proof of luck: An efficient blockchain consensus protocol. In: *Proceedings of 1st Workshop on system software for trusted execution* (p. 2).
40. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved September 27, 2019 from https://bitcoin.org/bitcoin.pdf.
41. Nguyen, G. T., & Kim, K. (2018). A survey about consensus algorithms used in blockchain. *Journal of Information Processing Systems*, *14*(1), 101–128.
42. Pass, R., & Shi, E. (2016). The sleepy model of consensus. Cryptology ePrint Archive, Report 2016/918. https://eprint.iacr.org/2016/918.
43. Pass, R., & Shi, E. (2018). Thunderella: Blockchains with optimistic instant confirmation. In: *Proceedings of annual international conference on the theory and applications of cryptographic techniques* (pp. 3–33).
44. Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, *22*(4), 299–319.
45. Schwartz, D., Youngs, N., Britto, A., et al. (2014). The ripple protocol consensus algorithm. Ripple Labs Inc White Paper (Vol. 5, p. 8).
46. Selimi, M., Kabbinale, A.R., Ali, A., Navarro, L., & Sathiaseelan, A. (2018). Towards blockchain-enabled wireless mesh networks. In: *Proceedings of 1st workshop on cryptocurrencies and blockchains for distributed systems* (pp. 13–18).

47. Sun, Y., Zhang, L., Feng, G., Yang, B., Cao, B., & Imran, M. A. (2019). Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment. *IEEE Internet of Things Journal*, *6*(3), 5791–5802.
48. Tanenbaum, A. S., & Steen, M. V. (2002). *Distributed systems: Principles and paradigms*. Beijing: Tsinghua University Press.
49. ThePiachu: Thoughts on delegated proof of stake and bitshares (2014). Retrieved September 27, 2019 from http://www.8btc.com/thoughts-ondelegated-proof-of-stake-and-bitshares.
50. Tromp, J. (2015). Cuckoo cycle: A memory bound graph-theoretic proof-of-work. In: *Proceedings of international conference on financial cryptography and data security* (pp. 49–62).
51. Wan, S., Gu, Z., & Ni, Q. (2019). Cognitive computing and wireless communications on the edge for healthcare service robots. *Computer Communications*. https://doi.org/10.1016/j.comcom.2019.10.012.
52. Wan, S., Li, X., Xue, Y., Lin, W., & Xu, X. (2019). Efficient computation offloading for internet of vehicles in edge computing-assisted 5g networks. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-019-03011-4.
53. Wang, C., & Jiang, H. (2015). Surf: A connectivity-based space filling curve construction algorithm in high genus 3d surface wsns. In: *Proceedings of 34th IEEE INFOCOM* (pp. 981–989). HongKong
54. Wang, C., Lin, H., & Jiang, H. (2014). Trajectory-based multidimensional outlier detection in wireless sensor networks using hidden markov models. *Wireless Networks*, *20*(8), 2409–2418.
55. Wang, C., Lin, H., & Jiang, H. (2016). CANS: Towards congestion-adaptive and small stretch emergency navigation with wireless sensor networks. *IEEE Transactions on Mobile Computing*, *15*(5), 1077–1089.
56. Wang, C., Lin, H., Zhang, R., & Jiang, H. (2017). Send: A situation-aware emergency navigation algorithm with sensor networks. *IEEE Transactions on Mobile Computing*, *16*(4), 1149–1162.
57. Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., et al. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, *7*, 22328–22370.
58. Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper (Vol. 151).
59. Wustrow, E., & Vandersloot, B. (2016). DDoSCoin: cryptocurrency with a malicious proof-of-work. In: *Proceedings of USENIX conference on offensive technologies* (pp. 168–177).
60. Xu, Y., Ren, J., Wang, G., Zhang, C., Yang, J., & Zhang, Y. (2019). A blockchain-based non-repudiation network computing service scheme for industrial iot. *IEEE Transactions on Industrial Informatics*, *15*(6), 3632–3641.
61. Xue, T., Yuan, Y., Ahmed, Z., Moniz, K., Cao, G., & Wang, C. (2018). Proof of contribution: A modification of proof of work to increase mining efficiency. In: *Proceedings of IEEE 42nd annual computer software and applications conference (COMPSAC)* (Vol. 1, pp. 636–644).
62. Yin, Y., Chen, L., Xu, Y., Wan, J., Zhang, H., & Mai, Z. (2019). Qos prediction for service recommendation with deep feature learning in edge computing environment. Mobile Networks and Applications. https://doi.org/10.1007/s11036-019-01241-7 (**to appear**).
63. Yin, Y., Yu, F., Xu, Y., Yu, L., & Mu, J. (2017). Network location-aware service recommendation with random walk in cyber-physical systems. *Sensors*, *17*(9), 2059.
64. Yuen, H.Y., Wu, F., Cai, W., Chan, H.C., Yan, Q., & Leung, V. (2019). Proof-of-play: A novel consensus model for blockchain-based peer-to-peer gaming system. In: *Proceedings of ACM international symposium on blockchain and secure critical infrastructure* (pp. 19–28).
65. Zaman, M.U., Shen, T., & Min, M. (2019). Proof of sincerity: A new lightweight consensus approach for mobile blockchains. In: *Proceedings of 16th IEEE annual consumer communications & networking conference (CCNC)* (pp. 1–4).
66. Zhao, P., Li, J., Zeng, F., Xiao, F., Wang, C., & Jiang, H. (2018). ILLIA: Enabling k-anonymity-based privacy preserving against location injection attacks in continuous lbs queries. *IEEE Internet of Things Journal*, *5*(2), 1033–1042.
67. Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In: *Proceedings of IEEE international congress on big data (BigData Congress)* (pp. 557–564).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Shaohua Wan received the joint Ph.D. degree from the School of Computer, Wuhan University and the Department of Electrical Engineering and Computer Science, Northwestern University, USA in 2010. Since 2015, he has been holding a post-doctoral position at the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology. From 2016 to 2017, he was a visiting professor at the Department of Electrical and Computer Engineering, Technical University of Munich, Germany. He is currently an associate professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His main research interests include deep learning for Internet of Things and edge computing. He is an author of over 80 peer-reviewed research papers and books. He is a senior member of IEEE.

Meijun Li received the B.E. degree from Wuhan University of Technology, China, in 2018. She is currently pursuing the M.S. degree in Electronics and Information Engineering at Huazhong University of Science and Technology, China. Her research interests include blockchain and Internet of Things.

Gaoyang Liu received the B.E. degree from Huazhong University of Science and Technology, China, in 2015. He is currently a Ph.D. candidate in School of Electronic Information and Communications at Huazhong University of Science and Technology. His research interests include security and machine learning.

Chen Wang received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow in the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently an associate professor. His research interests are in the broad areas of wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in wireless and mobile systems. He is a senior member of IEEE and ACM.