PoW-BC: A PoW Consensus Protocol Based on Block Compression

Bin Yu[1,2], Xiaofeng Li[1,2], and He Zhao[1*]
[1] Hefei Institutes of Physical Science, Chinese Academy of Sciences
Hefei 230031, China
[e-mail: yub@hfcas.ac.cn, xfli@hfcas.ac.cn]
[2] University of Science and Technology of China
Hefei 230026, China
[e-mail: zhaoh@hfcas.ac.cn]
*Corresponding author: He Zhao

Received June 11, 2020; revised December 7, 2020; accepted March 13, 2021; published April 30, 2021

Abstract

Proof-of-Work (PoW) is the first and still most common consensus protocol in blockchain. But it is costly and energy intensive, aiming at addressing these problems, we propose a consensus algorithm named Proof-of-Work-and-Block-Compression (PoW-BC). PoW-BC is an improvement of PoW to compress blocks and adjust consensus parameters. The algorithm is designed to encourage the reduction of block size, which improves transmission efficiency and reduces disk space for storing blocks. The transaction optimization model and block compression model are proposed to compress block data with a smaller compression ratio and less compression/ decompression duration. Block compression ratio is used to adjust mining difficulty and transaction count of PoW-BC consensus protocol according to the consensus parameters adjustment model. Through experiment and analysis, it shows that PoW-BC improves transaction throughput, and reduces block interval and energy consumption.

Keywords: Blockchain, Block Compression, Consensus, Mining Difficulty, Proof-of-Work

http://doi.org/10.3837/tiis.2021.04.011 **ISSN : 1976-7277**

1. Introduction

In recent years, blockchain technology has received more and more attention. Blockchain is a decentralized distributed ledger technology, and consensus can be reached without a third-party trust institution. Because of its characteristics of transparent, trustable, tamper-proof, traceable, and highly reliable, blockchain has shown a great application prospect in many industries [1-4].

“Consensus is the backbone of the blockchain and any other decentralized and distributed technology.” writed by C. Thompson [5]. There are different consensuses for different types of blockchains, such as proof-of-stake (PoS) [6, 7], Delegated proof-of-stake (DPoS) [8], Practical Byzantine Fault Tolerance (PBFT) [9, 10]. The first and still most common blockchain consensus protocol is proof-of-work (PoW) [11], which is adopted by Bitcoin [12, 13] to achieve consensus. Its core idea is to ensure the consistency of data and the security of consensus by introducing the computing power competition of distributed nodes [14]. But it has some drawbacks, such as large energy consumption and high cost. The miners do millions of computations per second, and PoW is costly and energy intensive [15]. To quote the MIT Technology Review: It's been estimated that Bitcoin guzzles about as much electricity annually as all of Nigeria [16].

Many innovative methods have been proposed to solve the problem of PoW consensus protocol. Y. Sompolinsky et al. present SPECTRE, which can operate at arbitrarily high block creation rates, and implies that its transactions confirm in mere seconds [17]. Z. Wang et al. propose a consensus protocol based on the credit model, which drew on the idea of personal credit risk assessment and a node credit model based on BP neural network was designed [18]. Y. Sompolinsky et al. address a security concern through the GHOST rule, a modification to the way Bitcoin nodes construct and re-organize the block chain, Bitcoin’s core distributed data-structure [19]. R. Zhang focuses on PoW consensus protocols, and extend a powerful method based on Markov decision processes to multiple utility functions [20].

Besides, there are also other consensus protocol which adopts PoW in the initial stage of blockchain operation. I. Bentov et al. propose proof-of-activity (PoA) [21], which builds upon the Bitcoin protocol by combining its PoW component with a PoS type of system. Proof-of-stake-velocity (PoSV) [22] is proposed as an alternative to PoW and PoS to secure the peer-to-peer (P2P) network and confirm transactions of Reddcoin, a cryptocurrency created specifically to facilitate social interactions in the digital age. K. Karantias et al. put forth the first cryptographic definition of what a proof-of-burn (PoB) [23] protocol is. It consists of a function which generates a cryptocurrency address and a verification function checks address. However, in these consensus protocols, PoW is only used to realize token allocation in the initial stage of blockchain.

On the other side, the scalability of block structure and size is being studying. Researchers at the Massachusetts Institute of Technology (MIT) have developed a cryptocurrency named Vault [24, 25], which allows a node to join the network by downloading only a fraction of total transaction data. The Bitcoin Core development team proposed BIP 152 (Bitcoin Improvement Proposal) for compact block relay [26]. With BIP 152, a transmitting node sends a compact block summary content to a node, and the data-receiving node uses the information received and the transactions in its memory pool to rebuild the entire block. Proposed by Andrew Clifford et al, the Xthin (extremely thin) block technology compresses block size to 1/24 of the original number of bytes [27]. Txilm Block presented by D. H. Ding et al, which is a type of lossy block compression with salted short hashing [28]. All the above work attempts to improve the scalability from the perspective of reducing the block size or improving the block data structure but are highly dependent on transactions in node's mempool. Sometimes, transmission times need to be increased.

This paper proposes an improved PoW consensus protocol: PoW-BC (proof-of-work-and-block-compression) consensus protocol. In this protocol, we put forward transaction optimization model and block compression model to reduce block size. Block compression ratio is an index to adjust mining difficulty and transaction count of PoW-BC consensus protocol according to the consensus parameters adjustment model.

The structure arrangement of this paper is as follows: The second part introduces overall architecture of PoW-BC consensus protocol and system architecture. The third part introduces transaction optimization model, block compression model and consensus parameters adjustment model. In the fourth part, PoW-BC consensus flow is descripted. In the fifth part, the experiment is performed adopting PoW-BC consensus protocol, and we analyze the compression ratio, compression and decompression duration, mining difficulty, transaction throughput, energy consumption and security. The sixth part summarizes this paper and outlines future work.

2. Overall Architecture

2.1 PoW-BC Consensus Protocol

PoW-BC consensus protocol is an improvement of PoW consensus protocol. The improvement points include transaction optimization, block compression, consensus parameters adjustment. This model is shown in **Fig. 1**.

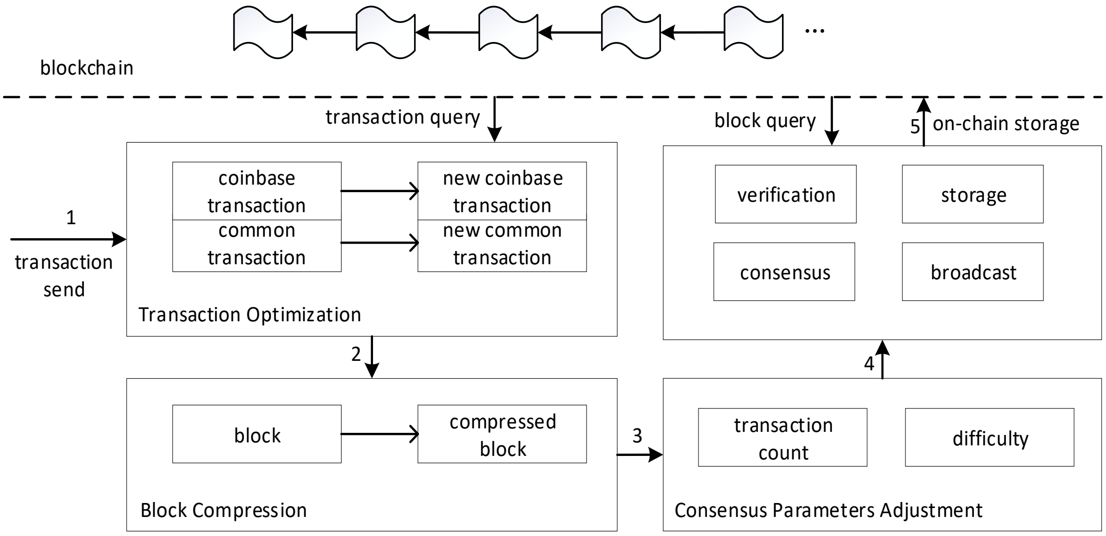


Fig. 1. PoW-BC consensus protocol

The transaction optimization model reduces the size of each transaction according to the data structure of coinbase transaction and common transaction. Furthermore, the block compression model makes the size of block smaller adopting an efficient data compression algorithm. And then, mining difficulty and transaction count of a block adjust to a new value according to the consensus parameters adjustment model. Every miner constructs a block using their own adjusted PoW-BC consensus parameters.

When a new block generates, it is broadcasted to every node in P2P network. Each node receiving a new block verifies this compressed block, and this new block is stored on the chain if it passes block verification.

2.2 System Architecture

The system architecture of a common blockchain system includes data layer, network layer, consensus layer, incentive layer, contract layer and application layer [29-31]. Data layer defines the data structure of blockchain, and includes distributed storage, digital signature, etc. Network layer includes P2P network, data transmission mechanism, data verification mechanism, etc. Consensus layer mainly achieves various consensus algorithms of network nodes. Incentive layer introduces economic factors into the blockchain, mainly including issuance mechanism and distribution mechanism of economic incentives. Contract layer is the basis of the programmable characteristics of blockchain. Application layer encapsulates various application scenarios and cases of blockchain. Among them, data layer, network layer and consensus layer are indispensable layers for a standard blockchain project.

The blockchain with PoW-BC consensus protocol improves almost all layers. Transaction optimization and block compression are adopted at the data layer. Any transaction and new block are verified by each block-receiving node at the network layer. When mining, miner adjusts mining difficulty and transaction count according to block compression rate. The system architecture is shown in Fig. 2.

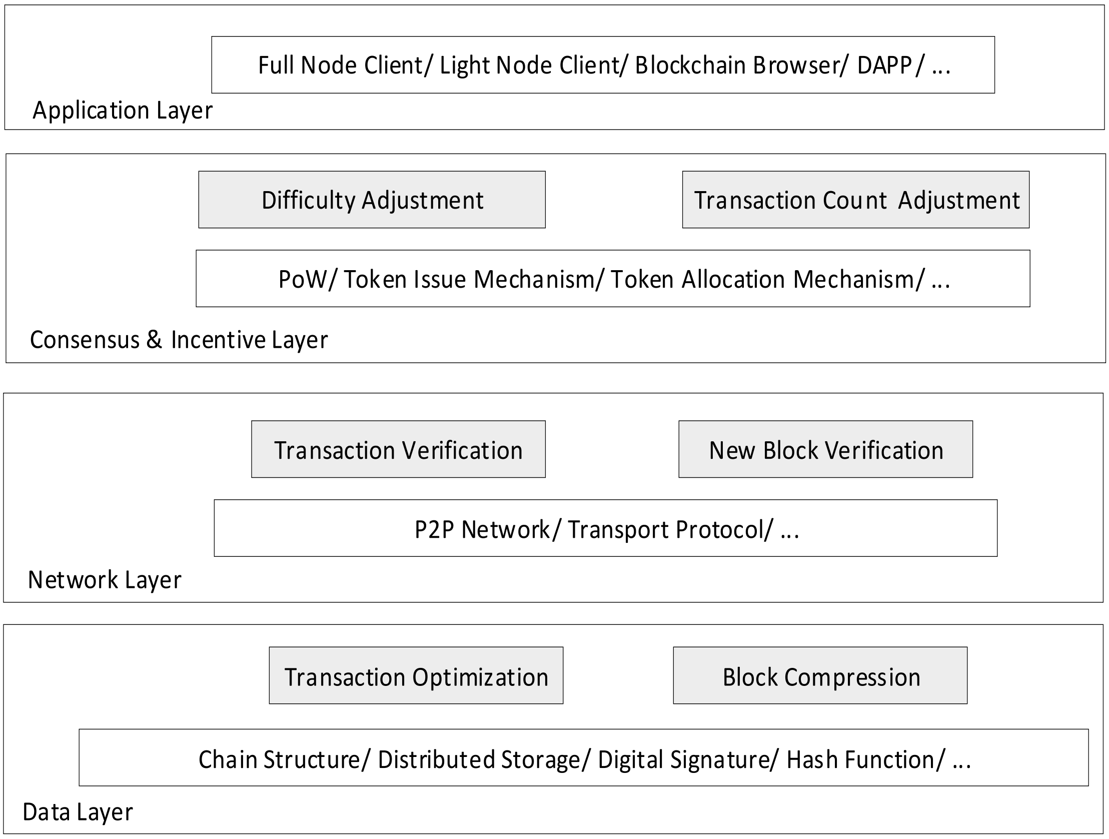

Full Node Client/ Light Node Client/ Blockchain Browser/ DAPP/ ...
Application Layer
Difficulty Adjustment
Transaction Count Adjustment
PoW/ Token Issue Mechanism/ Token Allocation Mechanism/ ...
Consensus & Incentive Layer
Transaction Verification
New Block Verification
P2P Network/ Transport Protocol/ ...
Network Layer
Transaction Optimization
Block Compression
Chain Structure/ Distributed Storage/ Digital Signature/ Hash Function/ ...
Data Layer

Fig. 2. The system architecture

There are the following main difficulties and challenges in the process of the system architecture:

1) Present or find an efficient data compression algorithm, which can compress block data with a lower compression ratio and less compression/ decompression time. Block compression model is described in detail in section 3.2.

2) Design a improved PoW consensus protocol, which address some shortcomings of PoW consensus protocol combined with block compression. We propose a PoW-BC consensus parameters adjustment model in section 3.3.

3) Implement a compressed block verification method. When receiving a new block, each node should verify compressed block data easily. This method is described in section 4.4.

3. Modelling

3.1 Transaction Optimization Model

In the Bitcoin blockchain, each block includes block size (4 Bytes), block header (80 Bytes), transaction counter (1 - 9 Bytes) and list of transactions. The size of first three items is very small, and it is hard to further optimize them. The number of transactions is very large, and each transaction contains a lot of information, part of which can easily get from the chain or is unnecessary. The size of transaction data reduces through optimizing transaction data structure.

There are two types of transaction: common transaction and coinbase transaction, which is always the first transaction appearing in every block. Two types of transaction are optimized respectively, and the optimization of coinbase transaction is shown in **Table 1**.

Table 1. The coinbase transaction optimization

Data Item	Size before optimization	Optimization	Size after optimization
Number of inputs	CompactSize	Always 0x01, and delete	0
Previous output hash	32 Bytes	All bits are zero, and delete	0
Previous output index	4 Bytes	All bits are ones: 0xFFFFFFFF, and delete	0

Number of inputs, previous output hash and previous output index of a coinbase transaction are all fixed values, and they are meaningless. So, we can delete them. The optimization of common transaction is shown in **Table 2**.

Table 2. The common transaction optimization

Data Item	Size before optimization	Optimization	Size after optimization
Previous output hash	32 Bytes	Replace with block height and transaction index	block height: 1 or 3 or 5 Bytes; transaction index: 1 or 3 Bytes
Previous output index	4 Bytes	Set to CompactSize type	1 or 3 Bytes

The size of previous output hash is 32 Bytes, and it can be replaced with the height of block, which includes previous output transaction, and transaction index. The size of previous output index is 4 Bytes, and it can be set to CompactSize [32] type. The size of a common transaction reduces by 25 - 33 Bytes.

When optimized transactions are verified, previous output hash can be queried from blockchain according to block height and transaction index.

3.2 Block Compression Model

The block compression model reduces the size of block adopting an efficient data compression algorithm. The benefits of block compression include: much faster transmission in P2P network, much smaller disk space for storing blocks. There are several definitions related to the block compression model.

1) Definition 1: Block Decompression

Block decompression is to recover compressed block into original block. The block after decompression should be the same as that before compression, so we adopt a lossless compression algorithm to compress and decompress block data.

2) Definition 2: Compression Ratio

Compression ratio is defined as the ratio between compressed data size and uncompressed data size. It is a measurement of the relative reduction in size of data representation produced by a data compression algorithm. Block compression ratio $R_{blockCompress}$ is expressed as the division of compressed block size $S_{blockCompress}$ by uncompressed block size S_{block}.

$$R_{blockCompress} = \frac{S_{blockCompress}}{S_{block}} \times 100\% \tag{1}$$

3) Definition 3: Compression and Decompression Duration

Compression and decompression duration are the amount of time we spend on the process of data compression and decompression. It is one of the important indexes to judge efficiency of compression algorithm. The shorter the compression and decompression time, the higher the compression efficiency.

4) Definition 4: Compression Algorithm

Compression algorithm is a method for reducing data size. There are many popular categories and types of compression algorithm, each of which works in a different manner, and some of which have results that differ in important ways. By combination of several compression algorithms, the size of compressed data is smaller.

The deflate algorithm [33] is a lossless data compression algorithm combining LZ77 algorithm [34] and Huffman coding [35]. It has a lower compression ratio and shorter compression/ decompression time compared with other algorithms, which includes gzip [36], lzw [37], zlib [38]. We adopt it as block compression algorithm in this paper.

3.3 Consensus Parameters Adjustment Model

In PoW consensus protocol, each miner has the same consensus parameters, and hash rate is one of the main bases for whether miners can construct a new block.

Each miner can self-adaptive adjust consensus parameters in PoW-BC consensus protocol according to block compression ratio. Consensus parameters include mining difficulty and transaction count of a block.

Because of transaction optimization and block compression, more transactions can be included in a block under the premise of block size limit of 1 MB, especially when the Bitcoin network is congested.

Mining difficulty is the relative amount of resources required to compete for constructing a new block. It is calculated dynamically at the end of roughly two-week epochs (or 2016-block periods) depending on whether the total estimated hash rate consumed by the network has increased or decreased. When hash rate of the whole network improves, mining difficulty increases to ensure that the average block interval is 10 minutes. A shorter block interval may result in security problems including increased fraud risks, more and longer forks [11, 12]. In this model, the lower the block compression rate, the easier it is to construct a block, namely smaller mining difficulty and block interval. Mining difficulty after adjustment $D_{adjusted}$ is:

$$D_{adjusted} = D_{base} \times \frac{T_{blockInterval}}{T_{baseBlockInterval}} \tag{2}$$

wherein D_{base} is mining difficulty before adjustment, $T_{baseBlockInterval}$ is block interval of the Bitcoin blockchain, and it is 10 minutes. $T_{blockInterval}$ is block interval after adjustment, and the expression is:

$$T_{blockInterval} = T_{const} + T_{transVerify} + T_{consensus} \tag{3}$$

wherein T_{const} is minimum block interval, and is a constant. $T_{consensus}$ is time for completing PoW-BC consensus. $T_{transVerify}$ is the sum of transmission time of new block broadcasted to the whole network and verification block time of receiving node. When $T_{transVerify}$ is 40s, more than 90% of nodes can receive block [39]. Stale block rate is not too large when block interval is 2.5 minutes [40], which is also block interval of Litecoin [41], so T_{const} is configured to 110s.

When a compressed block includes more transactions, more time is needed to verify transactions. The expression of $T_{transVerify}$ is:

$$T_{transVerify} = \begin{cases} 40, & T_{transVerify} < 40 \\ \frac{T_{baseTransVerify} \times S_{blockCompress}}{R_{blockCompress}}, & T_{transVerify} \geq 40 \end{cases} \tag{4}$$

wherein $T_{baseTransVerify}$ is the sum of transmission time and verification block time of receiving nodes when a new block of 1MB size is broadcasted to the whole network.

Consensus time is adjusted according to block compression ratio, and consensus time after adjustment $T_{consensusAdjusted}$ is:

$$T_{consensusAdjusted} = T_{consensus} \times R_{compressed} \tag{5}$$

wherein $T_{consensus}$ is consensus time before adjustment. According to (2), (3), and (5), $D_{adjusted}$ is expressed as:

$$D_{djusted} = D_{base} \times \frac{110 + T_{transVerify} + (490 - T_{transVerify}) \times R_{blockCompress}}{600} \tag{6}$$

$D_{djusted}$ is smaller than D_{base}, so it is easier to construct a block for miners after compressing block.

4. PoW-BC Consensus

The above three models are the basis of PoW-BC consensus protocol, and PoW-BC consensus flow is shown in **Fig. 3**.

Node receives this transaction after a user submits a transaction. The transaction is pre-processed by mining node and put into mempool (is where all the valid transactions wait to be confirmed by the Bitcoin network); The miner packages transactions and compresses block data, then constructs a new block after completing consensus. The transactions which has been packaged into this new block, should be deleted from mempool. After that, this new block is broadcasted to each node in P2P network; The block-receiving node verifies this new block.

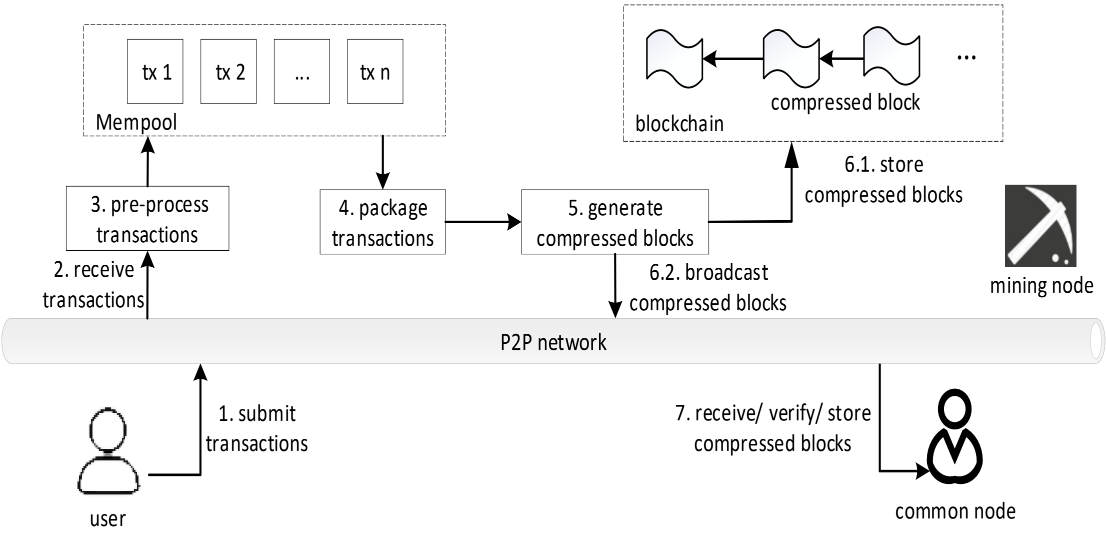

tx 1
tx 2
...
tx n
Mempool
compressed block
...
blockchain
3. pre-process transactions
4. package transactions
5. generate compressed blocks
6.1. store compressed blocks
2. receive transactions
6.2. broadcast compressed blocks
mining node
P2P network
1. submit transactions
7. receive/ verify/ store compressed blocks
user
common node

Fig. 3. The PoW-BC consensus flow

The process of PoW-BC consensus includes transaction pre-processing, transaction package, block generation, block verification and storage.

4.1 Transaction Pre-processing

Every transaction is broadcasted to each node. It is verified and optimized by each data-receiving node, and then it is put into mempool and waits for confirmation. Detailed steps are shown below:

1) Transaction verification

When a transaction is received, mining node verify the validity of transaction data. Each node saves a record of transactions in mempool, when a user submits a new transaction, it executes a sequence of checks to make sure the transaction is valid. The transaction signature script and public key script are checked by the Forth-like scripting system in Bitcoin blockchain, and transaction input is verified if public key script is true. This transaction is broadcasted to neighbor nodes of mining node after it is verified.

2) Transaction optimization

Transaction is further processed adopting transaction optimization model, which is described in detail in section 3.1. Then transaction is put into mempool to wait for confirmation.

Transaction pre-processing completes before PoW-BC consensus, so it does not increase the consensus time.

4.2 Transaction Package

When a new block is received, mining node starts to construct next block, and transactions waiting for confirmation in mempool need to be packaged. The transaction package and block compression are shown in **Fig. 4**.

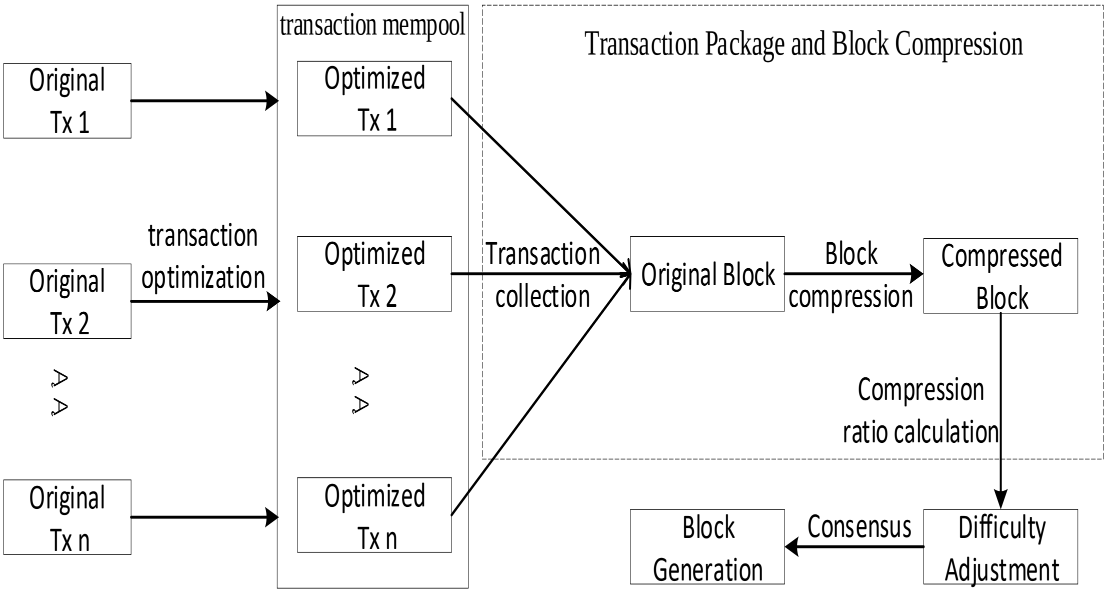


Fig. 4. The transaction package and block compression flow

The process of transaction package and block compression is shown below:

1) Transaction collection

Firstly, transactions, which have been packaged into previous block, should be delete from mempool. In general, miners are more willing to choose transactions with higher transaction fee and collect more transactions. Finally, the size of block including transactions cannot exceed 1MB.

2) Block compression

Now, block has included many transactions, but it doesn't have a correct nonce and timestamp. As described in section 3.2, block is compressed to a smaller size adopting a block compression algorithm.

3) Compression ratio calculation

Compression ratio, defined in section 3.2, is an important index in PoW-BC consensus protocol. Its value can be calculated according to (1). Because of block compression, more transactions can be packaged into a block.

4.3 Block Generation

Block generation includes mining difficulty adjustment, PoW-BC consensus and new block broadcasting. Mining difficulty is adjusted according to (4) and (6) in section 3.3, then miners start a mining until a correct nonce is calculated. When a new block generates, it is broadcasted to the P2P network of blockchain.

A detailed algorithm of the PoW-BC consensus process is shown in **Fig. 5**.

```
Algorithm: The PoW-BC consensus
Input:   rCompression:  The block compression ratio
Output:  result:        The result of this consensus
// Get adjusted mining difficulty according to (4) and (6).
dAdjusted ← GetDifficulty(rCompression)
// Calculate target hash by adjusted mining difficulty for comparing with block hash in mining.
hashTarget ← GetTargetHash(dAdjusted)
// Initialize the parameter value of nonce, timestamp and miningFlag.
nonce ← -1
timestamp ← GetNowTimestamp()
miningFlag ← false
// Start PoW consensus.
for i=0; i<maxNonce; i++
// Receive a message that another miner has completed a new block.
  receivingBlock ← SubscribeNewBlock()
// Verify the received new block. If passed verification, return and end this consensus. If not, continue this consensus.
  if receivingBlock != null && IsValidBlock(receivingBlock)
    return false
  end if
// Do hash calculation using a new value.
  nonce ← i
  timestamp ← GetNowTimestamp()
  hash ← GetBlockHash(blockData, nonce, timestamp)
// A nonce value that meets POW requirements is found.
  if hash <= hashTarget
    miningFlag ← true
    break
  end if
end for
// Do not find a nonce value that meets POW requirements.
if miningFlag == false
  return false
else
// Update block data using found nonce and timestamp.
  newblock ← UpdateBlock(blockData, nonce, timestamp)
// Broadcast this new block to nodes in p2p network.
  BroadcastBlock(newblock)
// Store this new block in local node.
  StoreBlock(newblock)
end if
return true
```

Fig. 5. The algorithm of PoW-BC consensus

If a mining node receives a new block constructed by another miner, it will stop this round of mining after this new block is verified.

4.4 Block Verification and Storage

Each new block is broadcasted to P2P network of blockchain, and verified by every block-receiving node. If a new block passes verification, it is stored on chain by node, and is broadcasted to neighbor nodes of this block-receiving node. A detailed algorithm of block verification and storage is shown in **Fig. 6**.

```
Algorithm: The block verification and storage

Input:   blockData:  The new block data
Output:  result:     The result of block verification
// Decompress block data adopting block compression model in section 3.2.
blockItem ← DecompressBlock(blockData)
// Verify mining difficulty using block compression ratio.
if IsValidDifficulty (blockItem.rCompress) == false
    return false
end if
// Verify block header according to POW consensus using nonce and other block data.
if IsValidBlockHeader(blockItem.header) == false
    return false
end if
// Verify all transactions in block.
for i=0; i< blockItem.TxCount; i++
// Verify each transaction.
    if IsValidTx(blockItem.Tx[i]) == false
        return false
    end if
end for
// Broadcast this block to nodes in p2p network.
BroadcastBlock(blockData)
// Store this block in local node.
StoreBlock(blockData)
return true
```

Fig. 6. The algorithm of block verification and storage

Compared with PoW, block verification and storage of PoW-BC has some more steps: firstly, compressed block should be decompressed, then mining difficulty is verified according to block compression ratio by forum (1). In the process of each transaction verification, previous output hash needs to be queried by block height and transaction index, which is described in section 3.1.

5. Experiment & Analysis

5.1 Experiment Design

We evaluate performance, security and efficiency of PoW-BC consensus protocol by experiment. We implemented a prototype of the PoW-BC consensus in GO programming language, consisting of approximately 8200 lines of code. We choose the deflate algorithm as the compression/ decompression algorithm for compressing and decompressing block data.

Bitcoin blocks of different heights are randomly chosen to demonstrate experimental data, and the experimental blocks is shown in **Table 3**.

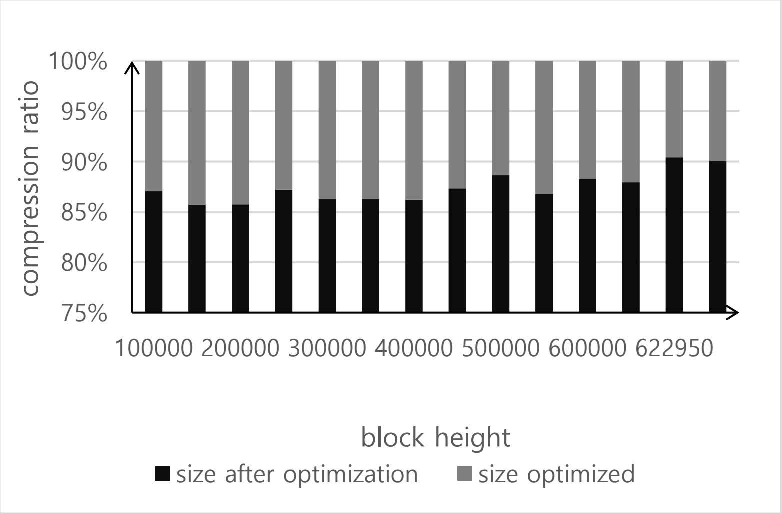
Table 3. The experimental blocks

Block Height	Transaction Count	Block Size (Bytes)	Mining Difficulty
100000	4	957	14484.16
100001	12	3308	14484.16
200000	388	247533	2864140.51
200001	32	11068	2864140.51
300000	237	128810	8000872135.97
300001	512	241334	8000872135.97
400000	1660	948994	163491654908.96
400001	1298	979159	163491654908.96
500000	2701	1048581	1873105475221.61
500001	2645	1066602	1873105475221.61
600000	1925	870371	13008091666971.9
600001	505	196097	13008091666971.9
622950	1590	1145300	13912524048945.9
622951	1507	1056590	13912524048945.9

In the experiment, we evaluate different blockchain parameters, such as compression ratio, compression and decompression duration, mining difficulty, throughput and energy consumption. Security is also analyzed.

5.2 Compression Ratio

Compression ratio is one of indexes to evaluate compression algorithm. The smaller the compression ratio, the more the data size is reduced. In the experiment, we compress randomly selected blocks adopting transaction optimization model and deflate compression algorithm, the analysis of the block compression ratio is shown in **Fig. 7**.



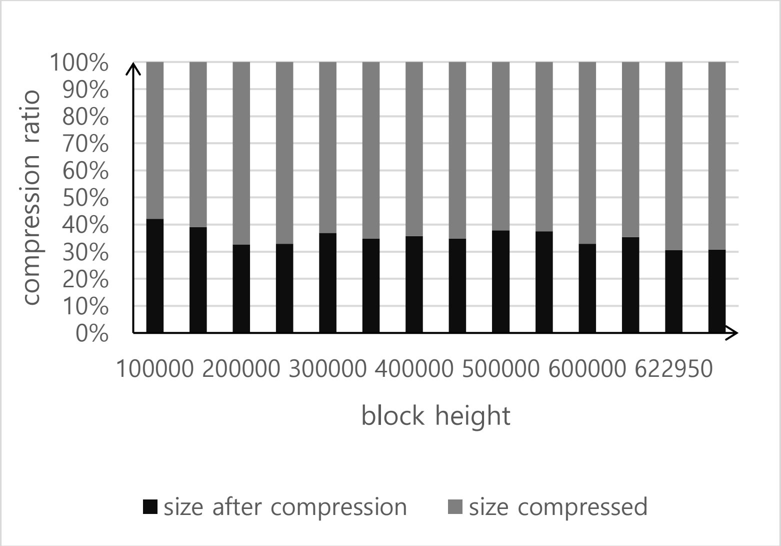


Fig. 7. The block compression ratio

Block compression ratio is between 85.73% and 90.43% after adopting transaction optimization model. Furth more, Block compression ratio is between 30.53% and 42.16% after adopting block compression model.

Transaction average compression ratio $R_{txCompress}$ is the ratio between the sum of all transactions compression rate to the total number of transactions in blocks. The equation is:

$$R_{txCompress} = \frac{\sum_{r=1}^{N_{block}} N_{tx}(r) \times R_{blockCompress}(r)}{\sum_{r=1}^{N_{block}} N_{tx}(r)} \times 100\% \tag{7}$$

wherein N_{tx} is the number of transactions in a block, and N_{block} is the number of blocks. The result of $R_{txCompress}$ is 34.81% in the experiment.

Each node disk space for storing blocks is smaller because of block compression. Block storage average compression ratio $R_{storageCompress}$ is the ratio between the size of all block data after compression and before compression. The equation is:

$$R_{storageCompress} = \frac{\sum_{r=1}^{N_{block}} S_{blockCompress}(r)}{\sum_{r=1}^{N_{block}} S_{block}(r)} \times 100\% \tag{8}$$

Namely, it is expressed as the division of all compressed blocks size by all uncompressed blocks size. The result of $R_{storageCompress}$ is 34.28% in the experiment. Up to Apr. 30, 2020, the blocks count of Bitcoin blockchain is 628220, and the blockchain size is 320GB [42]. Each full node only needs 110GB to store all blocks after adopting this model.

5.3 Compression and Decompression Duration

Compression duration is one of the most important indexes to evaluate the efficiency of compression algorithm. For mining node, shorter compression duration, higher efficiency. Compressed block needs to be decompressed for verifying by each node, so the shorter decompression duration, the better. The compression and decompression duration are shown in **Fig. 8**.

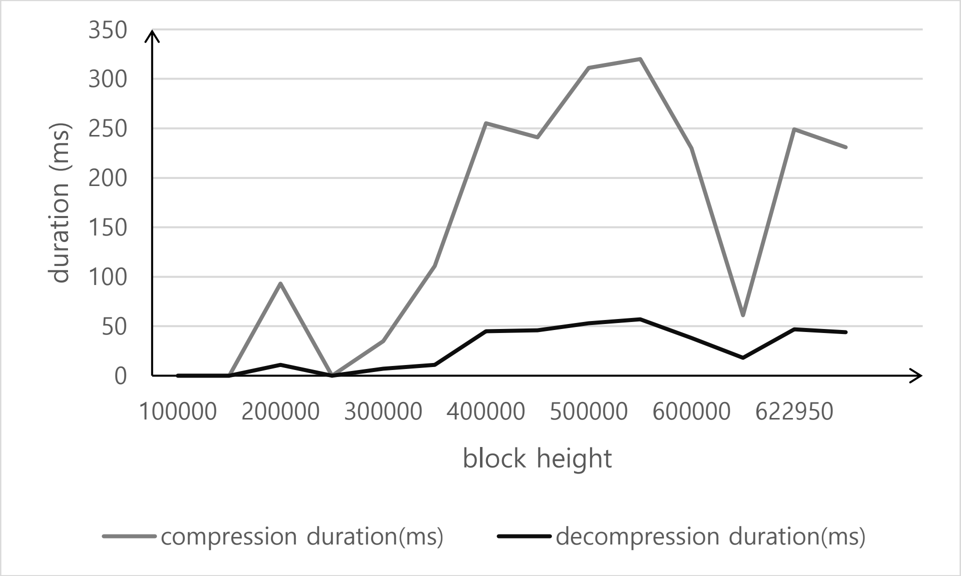
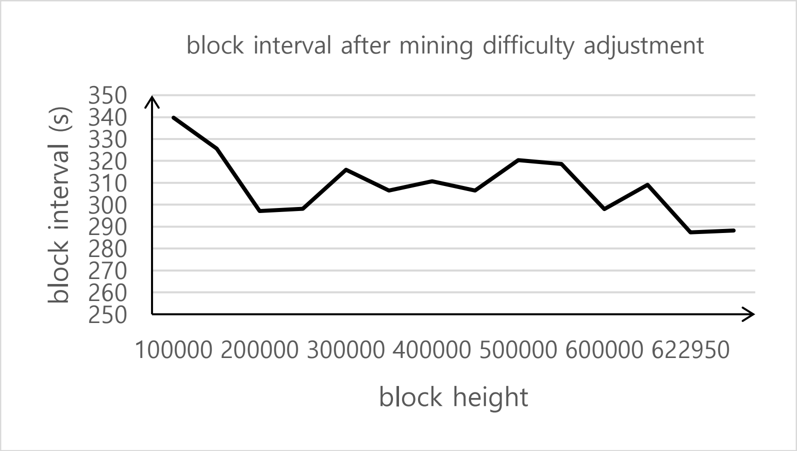


Fig. 8. The compression and decompression duration

Compression duration is between 0 and 320ms, and decompression duration is between 0 and 57ms. Compression duration is negligible compared with 10 minutes, which is the average block interval of the Bitcoin blockchain. Decompression duration is shorter, and it can be negligible.

5.4 Mining Difficulty

According to (2), mining difficulty is directly proportional to block interval. Block interval and mining difficulty adjustment ratio is shown in **Fig. 9**.



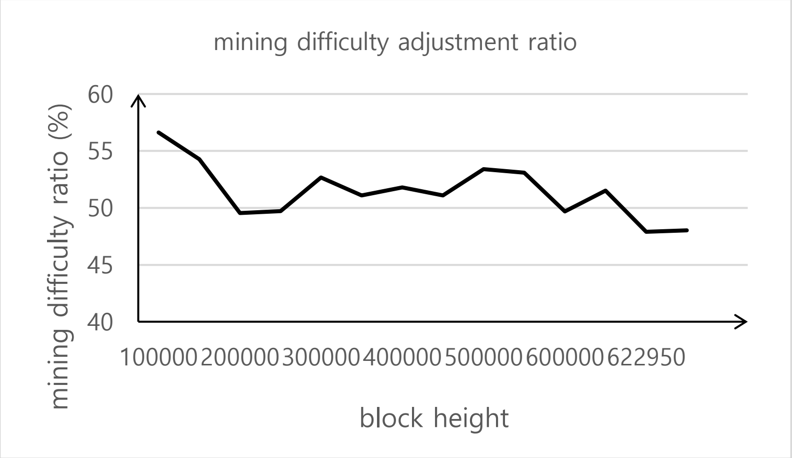


Fig. 9. The block interval and mining difficulty adjustment ratio

Block interval after mining difficulty adjustment is between 287.39s and 339.73s, and average block interval is 308.73s. Mining difficulty ratio after and before adjustment is between 47.90% and 56.62%, and average mining difficulty adjustment ratio is 51.46%. Block interval is smaller, then more blocks are generated in a certain period.

5.5 Transaction Throughput

Transaction throughput is one of the main indexes to measure the performance of a blockchain system, and it is represented as the number of transactions per second. The forum of transaction throughput N_{tps} is:

$$N_{tps} = \frac{S_{block}/S_{tx}}{T_{blockInterval}} \tag{9}$$

Before consensus parameters adjustment, average size of transaction S_{tx} is 250 Bytes [43], and average block interval $T_{blockInterval}$ is 10 minutes, so N_{tps} of Bitcoin blockchain is:

$$N_{tps} = \frac{1024\times1024/250}{600} = 6.99tps \approx 7tps \tag{10}$$

After consensus parameters adjustment, average transaction compression ratio $R_{txCompress}$ is 34.81%, and average block interval is 308.73s, so N_{tps} of Bitcoin blockchain is:

$$N_{tps} = \frac{1024\times1024/(250\times34.81\%)}{308.73} = 39.03tps \approx 39tps \tag{11}$$

Bitcoin's transaction throughput adopting PoW-BC is 5.6 times larger compared with PoW.

5.6 Energy Consumption

In PoW consensus protocol, all mining nodes in P2P network need to find a nonce by hash algorithm for generating a new block, and this process consumes a lot of power. The power consumption of constructing a block $N_{powerConsumption}$ is:

$$N_{powerConsumption} = \frac{N_{hashRate}\times N_{powerPerHash}}{N_{btcPerHour}} \tag{12}$$

wherein $N_{hashRate}$ is the sum of hash rate in the whole P2P network of Bitcoin blockchain, $N_{powerPerHash}$ is power consumption per hash calculation, $N_{btcPerHour}$ is the number of Bitcoin generated per hour.

According to the statistics of bitcoinblockhalf.com, up to Apr. 30, 2020, Bitcoins generated per day are 1800, and hash rate is 110.25Exahashes/s. Take the Antminer S19 Pro, produced by Bitmain Technology Holding Company, as an example， its hash rate is 110TH/s, and its power is 3250 W. The power consumption of constructing a block $N_{powerConsumption}$ is:

$$N_{powerConsumption} = \frac{110.25EH/s \times \frac{3250W}{110TH/s}}{(1800/24)/h} = 43432kWh \tag{13}$$

Adopting PoW-BC, the values of $N_{hashRate}$ and $N_{powerPerHash}$ do not change, but $N_{btcPerHour}$ becomes large because block interval become small. The average block interval is 308.73s according to mining difficulty analysis in section 5.4, so average block interval reduces to 51.46% (308.73/600), and power consumption also reduces to 51.46%.

Besides, transaction fee reduces because of transaction compression and the size of transaction become smaller. Generally, transaction fee is based on the byte size of a transaction: 0.0001BTC per 1KB (Calculated as 1KB when less than 1KB). Transaction average compression ratio is 34.81%, described in section 5.2, and transaction fee also reduces to 34.81%.

5.7 Security

Security is particularly important in blockchain systems, and there are many types of attacks in blockchain systems. We list several possible attack vectors and their solutions adopting PoW-BC consensus protocol.

1) Block Fork

PoW-BC consensus is more prone to block fork than PoW consensus, because block interval is shorter. As block fork of blockchain leads to resource waste, each node inquiries whether neighbor nodes receive a new block at a certain frequency in PoW-BC consensus protocol, in order to reduce the risk of block fork.

When block fork occurs, there are two status: if the height of block fork chains is different, the principle of taking the longest chain as the main chain is still followed. If the height of block fork chains is the same, the chain whose latest block with the smallest compression ratio is taken as the main chain.

2) 51% Attack

A 51% attack refers to an attack on a blockchain by a group of miners who control more than 50% of the network's mining hash rate. It is an attack most commonly on a blockchain adopting PoW, such as Bitcoin.

In PoW consensus protocol, if a group of miners control more than 50% of the network's mining hash rate, it can launch 51% attack. But in PoW-BC consensus protocol, even if a group of miners control more than 50% of the network's mining hash rate, it may not be able to launch 51% attack, because the number and order of transactions are different in a new block, which is packaged by each miner. Block compression ratio is not the same, which is a fine adjustment of mining difficulty, and weakens the weight of mining hash rate.

3) Block Compression Ratio Attack

When a miner constructs a new block, it uses a wrong or false block compression ratio. As described in section 4.4, each new block is verified by every block-receiving node. If a new block does not pass verification, it is not stored on chain by node, and is not broadcasted to the neighbor nodes of this block-receiving node.

4) Empty Block Attack

When a new bitcoin block is constructed, the miner can get some mining rewards. A miner can choose to include transactions in each block without any additional effort, and get fees attached with these transactions. Since there is no reason for miners to leave this additional income, miners will include transactions with every block.

In PoW-BC consensus protocol, compression ratio of empty block is not smaller than that of block including more transactions, so block interval is not affected by transactions count of a block.

5.8 Schemes Comparison

The decentralization, security, scalability, and energy consumption are fully considered at the time of design for this consensus. PoW, PoS and DPoS are main consensus schemes in blockchain projects, we compare PoW-BC with them as follow.

Comparison with PoW

1) Decentralization: The group of miners with large mining hash rate reduces the degree of PoW decentralization. PoW-BC consensus adjusts mining difficulty, and weakens the weight of mining hash rate.

2) Scalability: As adopting block compression, disk space for storing blocks is smaller, consensus period is less and transaction throughput is larger in PoW-BC scheme.

3) Security: PoW consensus offers superior security as high attack cost are required, and PoW-BC consensus follows high security of PoW.

4) Energy Consumption: Large energy consumption is main drawback in PoW consensus, power consumption also reduces with the decrease of mining difficulty in PoW-BC consensus.

5) Comparison with PoS and DPoS

In PoS and DPoS consensus, large energy consumption caused by mining is avoided, and consensus period is shortened. It requires lower performance of nodes in p2p network. But the degree of PoS and DPoS decentralization is reduced, and blockchain system is more prone to fork.

The transaction optimization and block compression model can also be applied to PoS and DPoS consensus. Stake value can be adjusted according to the parameter of the compression ratio, and it can improve the degree of PoS and DPoS decentralization.

6. Conclusion

PoW consensus protocol is commonly adopted in blockchain systems, but it has some shortcomings, such as large energy consumption and high cost. PoW-BC consensus protocol is designed to overcome these disadvantages of PoW.

In this paper, we present a transaction optimization model and block compression model to compress block data with a smaller compression ratio and less compression/ decompression duration. In consensus parameters adjustment model, block compression ratio is used to adjust PoW-BC consensus parameters to improve PoW consensus protocol. Through experiment and analysis, in the premise of ensuring blockchain secure and reliability, PoW-BC consensus protocol improves transaction throughput and transmission efficiency, reduces disk space for storing blocks, block interval and energy consumption.

The future work of this paper is to import other data compression algorithms, which has a smaller compression ratio to block data. And block compression model can be applied to PoS, DPoS, and other blockchain consensus protocol.

Acknowledgments

This study was supported with the National Natural Science Foundation of China (No. 61602435), Natural Science Foundation of Anhui Province (No. 1708085QF153).

References

[1] M. Pilkington, "Chapter 11: Blockchain technology: principles and applications," *Research Handbook on Digital Transformations, Research Handbooks in Business and Management series*, Sep. 2016, pp. 225-253. Article (CrossRef Link)

[2] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "BlockChain Technology: Beyond Bitcoin," *Applied Innovation Review*, no. 2, pp. 71, June 2016. Article (CrossRef Link)

[3] B. Yu, X. F. Li, and H. Zhao, "Structured Data Management Method Based on Scalable Blockchain Storage," *Transactions of Beijing Institute of Technology*, vol. 39, no. 11, pp. 1160-1166, Nov. 2019. Article (CrossRef Link)

[4] B. Yu, X. F. Li, and H. Zhao, "Virtual Block Group: A Scalable Blockchain Model with Partial Node Storage and Distributed Hash Table," *The Computer Journal*, vol. 63, no 10, pp. 1524-1536, May. 2020. Article (CrossRef Link)

[5] C. Thompson. "How does the Blockchain Work? (Part 2)," Aug. 2017. Article (CrossRef Link)

[6] W. T. Li, S. Andreina, J. Bohli, and G. Karame, "Securing Proof-of-Stake Blockchain Protocols," in *Proc. of European Symposium on Research in Computer Security International Workshop on Data Privacy Management Cryptocurrencies and Blockchain Technology*, California, pp. 297-315, 2017. Article (CrossRef Link)

[7] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," *PPCoinPaper*, pp. 1-6, Aug. 2012. Article (CrossRef Link)

[8] D. Larimer, "Delegated Proof-of-Stake (DPOS)," *Bitshare whitepaper*, 2014. Article (CrossRef Link)

[9] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proc. of the 3rd Symposium on Operating Systems Design and Implementation*, vol. 99, pp. 173-186, Feb. 1999. Article (CrossRef Link)

[10] H. Sukhwani, J. M. Martínez, X. L. Chang, K. S. Trivedi, and A. Rindos, "Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)," in *Proc. of 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pp. 253-255, Sep. 2017. Article (CrossRef Link)

[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Bitcoin*, 2008. Article (CrossRef Link)

[12] M. Andrychowicz, S. Dziembowski, D. Malinowski and L. Mazurek, "Secure Multiparty Computations on Bitcoin," in *Proc. of 2014 IEEE Symposium on Security and Privacy*, pp. 443-458, May 2014. Article (CrossRef Link)

[13] Bitcoin-Open Source P2P money, "Bitcoin is an innovative payment network and a new kind of money," Article (CrossRef Link)

[14] A. Li, X. H. Wei, and Z. He, "Robust Proof of Stake: A New Consensus Protocol for Sustainable Blockchain Systems," *Sustainability*, vol. 12, no. 7, Apr. 2020. Article (CrossRef Link)

[15] E. Maxie, "Pros and Cons of Different Blockchain Consensus Protocols," Mar. 2018. Article (CrossRef Link)

[16] M. Orcutt, "Blockchains Use Massive Amounts of Energy-But There's a Plan to Fix That," Nov.

2017. Article (CrossRef Link)

[17] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "SPECTRE: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections," 2017. Article (CrossRef Link)

[18] Z. Wang, Y. L. Tian, Q. X. Li and X. H. Yang, "Proof of work algorithm based on credit model," *Journal on Communications*, vol. 39, no 8, pp. 185-198, Sep. 2018. Article (CrossRef Link)

[19] Y. Sompolinsky and A. Zohar, "Secure High-Rate Transaction Processing in Bitcoin," in *Proc. of International Conference on Financial Cryptography and Data Security*, vol. 8975, pp. 507-527, July 2015. Article (CrossRef Link)

[20] R. Zhang, "Analyzing and Improving Proof-of-Work Consensus Protocols," Ph.D Thesis, Dept. of Electrical Engineering, University of Edinburgh, United Kingodm, 2019. Article (CrossRef Link)

[21] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake," *ACM Sigmetrics Performance Evaluation Review*, vol. 42, no 3, pp. 34-37, 2014. Article (CrossRef Link)

[22] L. Ren, "Proof of Stake Velocity: Building the Social Currency of the Digital Age," 2014. Article (CrossRef Link)

[23] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-Burn," in *Proc. of International Conference on Financial Cryptography and Data Security*, vol. 12059, pp. 523-540, 2019. Article (CrossRef Link)

[24] D. Leung, A. Suhl, Y. Gilad, and N. Zeldovich, "Vault: Fast Bootstrapping for the Algorand Cryptocurrency," in *Proc. of Network and Distributed Systems Security (NDSS) Symposium*, pp. 24-27, Feb. 2019. Article (CrossRef Link)

[25] MIT News Office, "A faster, more efficient cryptocurrency," Jan. 2019. Article (CrossRef Link)

[26] M. Corallo, "Compact Block Relay," 2016. Article (CrossRef Link)

[27] A. Clifford, P. R. Rizun, A. Suisani, A. Stone, and P. Tschipper, "Towards Massive On-Chain Scaling: Block Propagation Results with Xthin," 2016. Article (CrossRef Link)

[28] D. H. Ding, X. Jiang, J. P. Wang, H. Wang, X. B. Zhang, and Y. Sun, "Txilm: Lossy Block Compression with Salted Short Hashing," *zrXiv:1906.06500*, pp. 1-5, 2019. Article (CrossRef Link)

[29] R. Collins, "Blockchain: A New Architecture for Digital Content," *EContent: The magazine of electronic research & resources*, vol. 39, no 8, pp. 22-23, Nov.2016. Article (CrossRef Link)

[30] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things," *Security and Communication Networks*, vol. 9, no 18, pp. 5943-5964, Feb. 2017. Article (CrossRef Link)

[31] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Čapkun, "On the Security and Performance of Proof of Work Blockchains," in *Proc. of 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3-16, Oct. 2016. Article (CrossRef Link)

[32] "CompactSize," developer.bitcoin.org. Article (CrossRef Link)

[33] P. Deutsch, "DEFLATE Compressed Data Format Specification version 1.3," *Internet Engineering Task Force(IETF)*, May 1996. Article (CrossRef Link)

[34] "The LZ77 algorithm," Data Compression Reference Center: RASIP working group. Faculty of Electrical Engineering and Computing, University of Zagreb, Jan. 2013. Article (CrossRef Link)

[35] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, vol. 40, no 9, pp. 1098-1101, Sep. 1952. Article (CrossRef Link)

[36] "GNU Gzip: General file (de)compression," Free Software Foundation Inc., 2009-2018. Article (CrossRef Link)

[37] M. Dipperstein, "Lempel-Ziv-Welch (LZW) Encoding Discussion and Implementation," dipperstein.com, Mar. 2015. Article (CrossRef Link)

[38] J. Gailly and M. Adler, "zlib compression and de compression," zlib.net. Article (CrossRef Link)

[39] X. D. Chen, "Why Bitcoin block interval is 10 minutes," Feb. 2019. Article (CrossRef Link)

[40] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," in *Proc. of ACM SIGSAC Conference on Computer and Communications Security*, pp. 3-16, Oct. 2016. Article (CrossRef Link)

[41] "Litcoin Home Page," litecoin.org. Article (CrossRef Link)
[42] "Bitcoin, Litecoin, Namecoin, Dogecoin, Peercoin, Ethereum stats," bitinfocharts.com. Article (CrossRef Link)
[43] H. Hossein, H. Xu, and S. Emmanuel, "Big-Crypto: Big Data, Blockchain and Cryptocurrency," Big Data Cogn. Comput, vol. 2, no. 4, Oct. 2018. Article (CrossRef Link)

Bin Yu received the B.S. degree in mechanical engineering from Anhui Polytechnic University, Wuhu, China, in 2004, and the M.S. degree in mechanical and automotive engineering from Hefei University of Technology, China, in 2007. He is currently pursuing a PhD of computer applied technology at University of Science and Technology of China, Hefei, China. His research interests focus on blockchain technology and blockchain scalability.

Xiaofeng Li received the B.S. degree from Tianjin University in 1987. He is currently a research professor of Hefei Institutes of Physical Science, Chinese Academy of Sciences (CASHIPS), and a doctoral supervisor at the University of Science and Technology of China. He is the director of Internet network information center of CASHIPS, vice chairman of Hefei branch of Association for Computing Machinery (ACM), and vice chairman of Anhui Radio Technology Association, etc. His current research interests focus on blockchain technology, computer applied technology and measurement and control technology and automation instrument.

He Zhao received the PhD degree from the University of Science and Technology of China in 2016, and B.S. and M.S. degrees from Nanjing University of Posts and Telecommunications in 2007 and 2010 respectively. He has been working for Huawei Technologies from 2010 to 2011. He is currently a senior engineer and the director of Network Information Center at Hefei Institutes of Physical Science, Chinese Academy of Sciences. His research interests include computer networking, health informatics, blockchain technology and software architecture.