IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 30 May 2023, accepted 4 June 2023, date of publication 7 June 2023, date of current version 16 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283928

RESEARCH ARTICLE

Cooperative Mining System to Improve Bitcoin Scalability

DALIA ELWI[1], OSAMA ABU-ELNASR[1], AHMED S. TOLBA[1,2], AND SAMIR ELMOUGY[1]
[1]Computer Science Department, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt
[2]New Heliopolis Engineering Institute, Cairo 11829, Egypt

Corresponding author: Dalia Elwi (dalia_elwi@mans.edu.eg)

ABSTRACT Blockchain is a Distributed Ledger Technology (DLT) that allows users to exchange values directly without a need for trusted third parties. Bitcoin is one of the most popular digital cryptocurrencies that is based on blockchain technology. However, it faces scalability problems including transaction throughput, latency, and starvation. Bitcoin transaction throughput is very low compared to traditional payment methods. Additionally, many Bitcoin transactions suffer from delays and starvation as miners prefer transactions with higher fees. More of the current research focuses on how to enhance Bitcoin scalability by improving the performance of consensus techniques, dividing the network into smaller ones with different parts of the blockchain, or completely changing the blockchain data structure. Unfortunately, engaging in this problem usually affects either decentralization or security; this is called the blockchain trilemma. This paper proposes the Cooperative Mining System (CMS), which depends on enhanced proof of work consensus algorithm. This proposed system increases transaction throughput and eliminates transaction latency and starvation without affecting decentralization and security. In CMS for each epoch, miners cooperated to create one super-block that contains more transactions than the traditional Bitcoin block. Whereas miners create their traditional blocks simultaneously, broadcast them, wait to receive other miners' blocks, and lastly create a super-block that contains all the transactions of the gathered blocks. The Simulation results of the CMS and Bitcoin system using different case scenarios show a significant improvement in CMS compared to the current Bitcoin system. The CMS greatly increases the transaction throughput and eliminates transaction latency and starvation.

INDEX TERMS Bitcoin, blockchain, blockchain trilemma, consensus algorithm, cooperative mining, cryptocurrency, latency, proof of work, scalability, throughput.

I. INTRODUCTION

Recently, the field of blockchain has attracted the attention of many researchers due to its features that distinguish it from other storage systems. Blockchain is a decentralized, cryptographically secured, and scalable ledger that can store data in a linked list of blocks. It has many applications, but the most important of them is the digital cryptocurrencies such as Bitcoin [1], [2], Ethereum [3], IOTA [4], Ripple [5], Bitcoin Cash [6], Cardano [7], Litecoin [8], Monero [9], Neo [10], and Dash [11]. According to the Coin Market Cap ranks, Bitcoin is the first and most popular among all other cryptocurrencies [12]. All the cryptocurrencies based on blockchain technology are facing the problem of blockchain trilemma [13]. It means that the blockchain system can only optimize two of the three objectives: decentralization, scalability, and security, at the same time. The solution for this trilemma is still under research [13]. Because the most important cryptocurrency to date is Bitcoin, we are focusing on enhancing its scalability to achieve high performance without affecting other objectives. Scalability is a problem that is divided into three subproblems [14]:

A. LIMITED TRANSACTION THROUGHPUT

Bitcoin processes far fewer transactions per second than traditional methods. It verifies only 17 transactions per second,

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

but Visa can process thousands of transactions per second as an example of a traditional payment method [14]. It cannot fulfill the requirements of processing millions of transactions in real life.

B. TRANSACTION LATENCY AND STARVATION

Block propagation delay is affected by the block size. A large block causes a significant delay in block propagation [15]. In Bitcoin, one block is mined every 10 minutes, and the capacity of blocks is very small (on average 1MB for each block) [16]. Therefore, many small transactions with low fees might be delayed since miners prefer those transactions with high fees. This problem is called transaction latency. Also, some transactions with low fees suffer from starvation [17]. They will never be picked up by any miner, especially if the number of waiting transactions is huge.

C. STORAGE LIMITATION

Miners in Bitcoin need to store the complete blockchain to be able to retrieve historical activities, search transactions, and validate new transactions [18]. Over time, the size of the blockchain will be huge. Consequently, miners with limited storage capacities will quit. Therefore, the blockchain network will be more centralized.

In this paper, we focus on solving the transaction throughput, latency, and starvation limitations in Bitcoin. This paper is organized as follows. Section II discusses some background about the Bitcoin blockchain. Section III illustrates some of the related works. Section IV explains the proposed CMS system. Section V discusses the implementation and evaluation results of the proposed CMS system. Lastly, section VI summarizes the conclusion and future work.

II. BACKGROUND

A. BITCOIN

Bitcoin ensures decentralization by using a Peer-to-Peer (P2P) network. Besides, it uses a digital signature and cryptographic hash functions to guarantee a high level of security [19]. A digital signature is created by Elliptic Curve Digital Signature Algorithm (ECDSA) [20], [21]. It verifies the transaction and ensures that the sender has the full right to spend the associated funds. A Hash function is a one-way cryptographic function that is used to generate fixed-length output data for any size of source data. Bitcoin uses the SHA-256 hash algorithm to hash blocks and transactions. Although Bitcoin guarantees both decentralization and high security, it suffers from scalability problems [13]. In Bitcoin, users broadcast their transactions to network nodes which are called miners. Then miners choose transactions with high fees to be validated and add them to a new block every 10 minutes. So, transactions with low fees are exposed to the problem of starvation [22].

B. DIGITAL SIGNATURE

Transactions are based on two types of keys: *Private* and *Public* keys. The private key is a 256-bit binary number, while the public key is a unique number that is calculated from the private key. SHA256 algorithm is used to generate the private keys in a random manner [23]. The elliptic Curve Multiplication function is a one-way function that is used to calculate the public keys [24]. The private keys are unknown to the network. Therefore, a digital signature is used as evidence of a connection between public and private keys [25]. It is used in the transaction authentication process, which consists of two steps: *Signing* and *Verifying*. A digital signature is signed on the transaction data in the signing step. On the other hand, verifying step proves that the digital signature and the public key were generated from the same private key [26].

C. CONSENSUS ALGORITHM

Each miner in the Bitcoin network has its own replica of the blockchain. So, to reach consistency, a consensus algorithm should be used. Various types of consensus algorithms had been developed with different mechanisms such as Proof of Work (PoW) [27], Proof of Stake (PoS) [28], Delegated Proof of Stake (DPoS) [29], and Practical Byzantine Fault Tolerance (PBFT) [30]. Table 1 presents a comparison between these four consensus algorithms based on their representative cryptocurrencies and the blockchain trilemma in them.

TABLE 1. Comparison of consensus algorithms [13], [31].

Consensus Algorithm	Crypto Currency	Scalability	Security	Decentralization
PoW	Bitcoin Ethereum Litecoin Monero Zcash	Low	High	High
PoS	Nano Nxt Qtum	Low	Medium	High
DPoS	EOS Cardano TRON Lisk BitShares	Medium	Medium	Low
PBFT	Ripple Stellar Zilliqa	Medium	Medium	Medium

D. PROOF OF WORK

PoW is one of the reasons for the popularity and success of Bitcoin because of [32]:

- It maintains the block creation time at roughly 10 minutes to ensure that blocks are more secure and immutable.
- It prevents or at least mitigates the occurrence of double spending (spending the same coins in two or more transactions) [32].
- It avoids problems caused by network propagation delay.
- It avoids hard forks which occur if a miner still propagates its block while another miner, who hasn't yet received this block, creates and starts to broadcast its own block of the same height [33].

So, all miners are forced to wait these 10 minutes via competition to solve a puzzle. This competition is called the mining process. The miner who solves the puzzle first is the winner and then it can create a new block and take the rewards. Each block has a hash value that represents its data including transactions, timestamp, nonce, difficulty, and previous block hash [1] as shown in Figure 1. Bitcoin uses the SHA-256 algorithm to calculate hash values. So, simply, the puzzle is calculating the hash number of the new block continuously until reaching a certain condition. In each iteration, the miners increase the nonce number by 1. For example, the hash number is correct if it begins with five zeros or if it is less than a specific target threshold that is calculated based on the difficulty level [34].

The difficulty is a measure of how difficult the mining is [34]. In Bitcoin [2], difficulty should be adjusted every 2016 block (2 weeks), so that the time between each block remains 10 minutes. If the hash rate is high, then the miners can reach the correct hash in a much shorter time and therefore, the difficulty should be high. And vice versa, if the hash rate is low, then the miners can reach the correct hash in a much longer time and therefore, the difficulty should be low. The hash rate is related to the total mining power that is used by all miners in the network [35]. In other words, it depends on the network size and the processing capabilities of the miners.

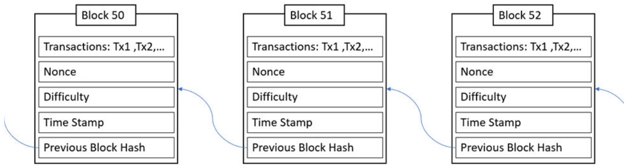

FIGURE 1. Block structure.

If the difficulty is low, more than one miner could create a block at the same time then forks appear. The longest chain rule should be applied to the next blocks in this case. This rule means that the new blocks are added to the longest fork which consumes the most energy to be built. The other chains are ignored, and their transactions are returned to the memory pool (*MemPool*), where transactions are waiting for confirmation. The blocks in these chains are called orphans as shown in Figure 2 [36].

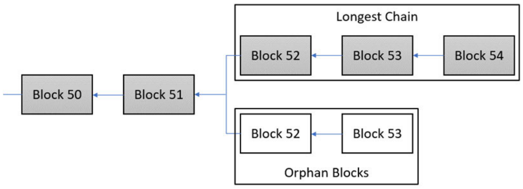

FIGURE 2. Longest chain rule.

III. RELATED WORK

Many researchers focus on solving the scalability problem of cryptocurrencies, especially in Bitcoin, by enhancing the performance of the consensus algorithm. As proposed in [37], Shihab and Qusay accelerated the process of proof of work based on parallel mining instead of solo mining. They relied on making sure that no miners put the same effort into solving the same specific block. In this method, all miners try to create the same block with the same transaction data but with different nonce groups which are distributed among them by the manager. Their results showed 34% improvements in scalability compared to the traditional Bitcoin system. This method inadvertently shortened the block creation time. Therefore, a lot of forks will appear, and the consensus will not be guaranteed. It also needs more trust in the manager and thus leads to more centralization.

Directed Acyclic Graph (DAG) [38] based solutions are also suggested like Conflux [39], and IOTA [4]. Chenxing et al. proposed a Conflux [39] system to create concurrent blocks to increase the throughput. Conflux can achieve 2.88GB/h transaction throughputs when 20K nodes are running. In Conflux, the data structure that is used to represent the blockchain is DAG rather than the linked list. Also, it no longer depends on consensus protocols. This method assumes that no two concurrent blocks have conflicting transactions and that only a ‘‘happens-before’’ relationship exists between blocks. There are no forks in this method and the order of blocks is determined according to the edges between blocks which are parent edges and reference edges. Their results showed that this work achieves high throughput, but it negatively affected the security and the blockchain has become more vulnerable to attacks.

Another DAG-based, parallel, and distributed ledger is called Tangle. Serguei [40] proposed a cryptocurrency called IOTA, that uses tangle technology and is specially designed for the Internet of Things. There are no miners and transaction fees in IOTA. The user who wants to make a transaction should approve two previous transactions from other users. So, it is more scalable than blockchain-based cryptocurrencies. At the same time, the IOTA network is a bit centralized and less secure than Bitcoin. That is because all the transactions are verified by coordinator nodes.

Increasing the block size is an alternate solution to the Bitcoin scalability problem as proposed in Bitcoin-NG [41], SegWit [42], Bitcoin Cash [6], and Bitcoin Classic [43]. In Bitcoin-NG [41], Ittay et al. improve the performance of Bitcoin by splitting the mining process into two operations: leader election and transaction serialization. The time of mining is divided into epochs and in each, there is only one leader who is selected randomly. A leader is responsible for adding transactions into the blocks until the next epoch comes. There are two types of blocks: microblocks and keyblocks. Microblocks are created by leaders, and they contain transactions. keyblocks contain information about the leaders of each epoch. This method is more vulnerable to the single-point-of-failure problem where each epoch is controlled only by one leader. Therefore, decentralization is reduced.

The Bitcoin network upgrade, which is called SegWit [42], increases the transaction throughput by enlarging the size of the block from 1MB to 4 MB. The drawback of this upgrade is that the large block takes a long time to reach every miner in the network causing hard forks.

On the other hand, the sharding technique is used in some research as [44], [45], [46], [47], and [48] to scale the blockchain systems by dividing the network into small parts, which are called shards. Each shard contains a group of nodes and can create blocks in parallel, thus increasing the transaction throughput. Some of the challenges of this method are the way of maintaining security and decentralization, and how to reach a consensus in the entire network. Despite the scalability improvements, sharding technologies are compromising the security of blockchain which makes the single shard vulnerable takeover attack [14] possible.

Table 2 summarizes the difference between the proposed CMS system and the related works previously mentioned.

TABLE 2. The difference between the proposed CMS system and the related works.

System	Technique	Security	Decentralization	Scalability
[37]	Enhanced consensus algorithm	Affected	Affected	Improved
[39]	DAG	Affected	Not affected	Improved
[40]	DAG	Affected	Affected	Improved
[41]	Increasing the block size	Affected	Affected	Improved
[42]	Increasing the block size	Affected	Not affected	Improved
[44-48]	Sharding	Affected	Affected	Improved
The proposed CMS	Enhanced consensus algorithm	Not affected	Not affected	Improved

IV. PROPOSED SOLUTION

Miners in the Bitcoin system are competing by performing a Proof of Work [27]. Only the winner is the one who creates the new block with an average size of 1MB and broadcasts it to the rest of the network. In contrast, this paper proposed Cooperative Mining System (CMS) that allows miners to be cooperative but independent. All miners create new blocks of the same size as in Bitcoin and broadcast them to the rest of the network. Then each miner individually generates one super-block containing all the transactions of the blocks that have been received. The super-block is not broadcasted over the network, but the consensus is reached using PoW. In addition to all the features that PoW provides as we mentioned previously, it forces all miners to wait for a while until they can aggregate other blocks that have been broadcast. This period is controlled by the difficulty level and the network size as in traditional Bitcoin. At the end of each epoch, all miners will have the same replica of the super-block.

Miners in CMS randomly choose the transactions to create their new blocks. So, the super-block will contain all the transactions in these blocks, but the redundant transactions are removed. The size of the super-block varies depending on the number of received blocks and the number of redundant transactions. After each epoch, if M miners exist in the network, a block can contain N transactions, there are no ignored blocks, and all blocks created by all miners are integrated:

- The maximum number of transactions in the super-block is $M \times N$ if each miner chooses different transactions for its block. Whereas all blocks created by all the miners have different transactions and there are no redundant transactions.
- The minimum number of transactions in the super-block is N if all miners choose the same transactions for its block. Whereas all blocks created by all the miners have the same transactions.so, the redundant transactions are removed.

The following inequality presents the maximum and minimum number of transactions in the super-block.

$$N \leq Superblock\ Transactions \leq M \times N$$

Figure 3 and Figure 4 show the difference between the competition in the Bitcoin mining system and the cooperation in the proposed CMS mining system.

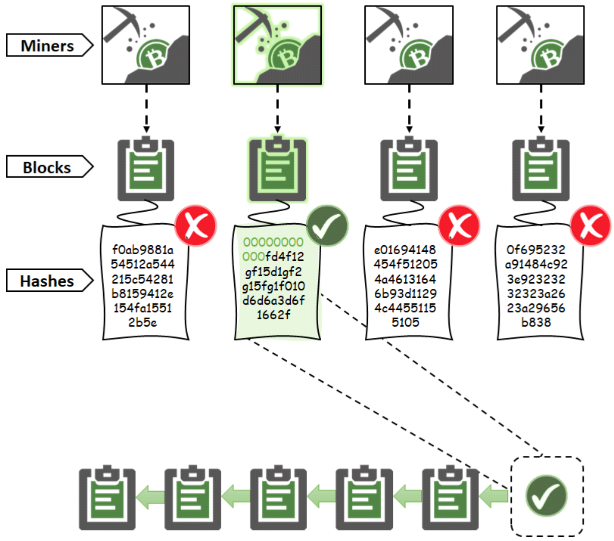

FIGURE 3. Competition in the Bitcoin mining system.

A. COOPERATIVE MINING SYSTEM

The proposed CMS is divided into four stages: initialization, waiting, integration, and rewarding, as illustrated in Figure 5.

1) INITIALIZATION

At this stage, miners individually select several random transactions from the *MemPool*, depending on the block size, to create their new blocks. Unlike Bitcoin, miners create their

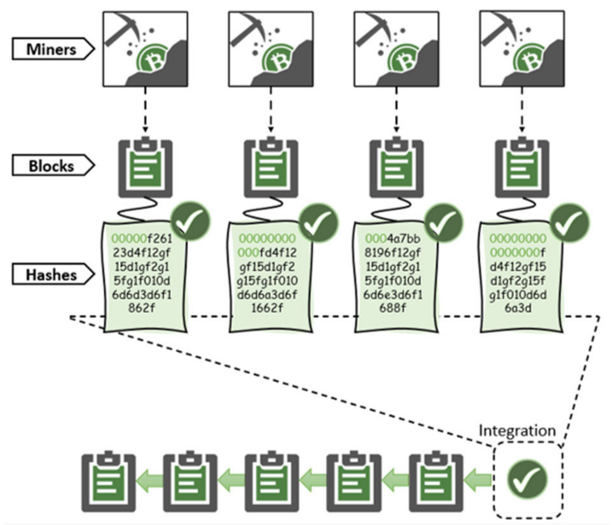

FIGURE 4. Cooperation in the proposed mining system (CMS).

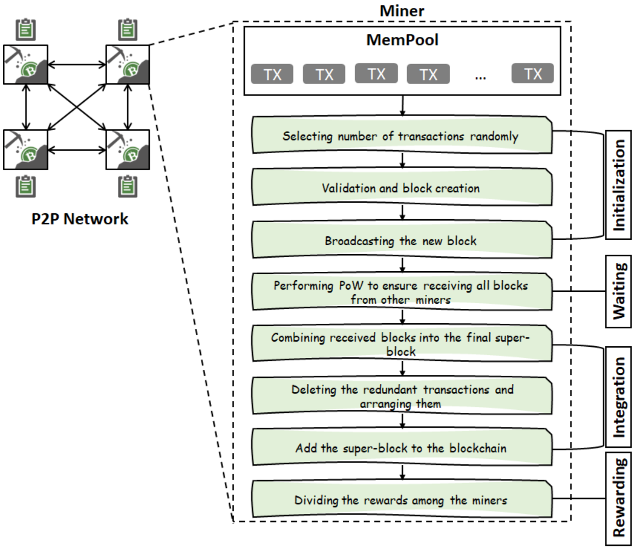
P2P Network
Miner
MemPool
TX
TX
TX
TX
...
TX
Selecting number of transactions randomly
Validation and block creation
Broadcasting the new block
Performing PoW to ensure receiving all blocks from other miners
Combining received blocks into the final super-block
Deleting the redundant transactions and arranging them
Add the super-block to the blockchain
Dividing the rewards among the miners
Initialization
Waiting
Integration
Rewarding

FIGURE 5. The proposed CMS system.

blocks without performing PoW and then broadcast those blocks over the network. Therefore, there is no competition in the proposed CMS system. All blocks are useful and will be used to create the final super-block.

2) WAITING

After broadcasting, each miner implements PoW to ensure its previously mentioned features. Besides, it forces all miners to wait until they can receive all blocks from other miners. As in the Bitcoin system, the waiting duration is controlled by the difficulty level [49] which changes based on the number of network miners, network bandwidth, and miners' hash rates.

3) INTEGRATION

At this point, one large super-block is created by each miner, individually. It contains all the transactions in the received blocks, without repetition, arranged by their hash numbers. Therefore, consistency is achieved, and all the miners obtain the same copy of the blockchain.

4) REWARDING

All miners create blocks, broadcast them, implement PoW, and create super-blocks. So, a certain amount of cryptocurrency currently is 6.25 in Bitcoin, as well as the transaction fees will be divided equally among all miners in the network. The throughput of transactions will be increased, and consequently, the revenues of miners will increase.

B. SECURITY ASPECTS

The proposed CMS system can solve scalability problems by increasing the transaction throughput and eliminating transaction latency. At the same time, it is proposed to maintain the decentralization and security levels as in the traditional Bitcoin. The proposed CMS system guarantees all security aspects of the traditional Bitcoin such as the following:

1) MINER MALFUNCTION

If the miner has any kind of malfunction and is unable to send the block to the rest of the miners, miners just wait for a time that is determined by the difficulty level while ignoring the disrupted miner. Next, the rewards are divided among the miners involved in the mining process whereas the inoperative miners are usually ignored. When the disrupted miners come back to the network, they reload the blockchain from any other active miners and return to participate in the mining process.

2) BLOCK LOSS

In the case of block loss, while broadcasting it over the network, forks appear, and miners create different copies of the super-block. Miners can resolve this problem in the next epoch. During the creation of the next block, if the miner creates a block containing a previous hash that is different from most of the received blocks, then it reconnects to the network and reloads the blockchain from these miners and its block is ignored.

3) BLOCKCHAIN ALTERING

Since each block is linked to the previous one by using its hash, it is very costly to alter any block of the blockchain. Any changes in the block cause changes in its hash. So, it needs to mine all the following blocks to maintain the connection between these blocks. The structure of the blockchain and using PoW as a consensus algorithm for mining make the blockchain immutable.

4) DOUBLE SPENDING

The same units of the currency can be spent twice in two separate transactions. The proposed CMS solves this problem, like Bitcoin, using distributed blockchain to store the transactions and using the PoW consensus algorithm for mining. If two transactions spend the same tokens, one of them will be valid and the other is invalid. Also, if the two transactions are added to two new blocks at the same time, only the longest chain is valid and the other is ignored.

5) SELFISH MINING

Miners can hide the newly created blocks and introduce them later to alter the blockchain and gain more rewards. The proposed CMS solves this problem since each super-block creation depends on all the miners in the network not only one miner that is afraid to be selfish.

C. SCALABILITY METRICS

The system measures scalability in two aspects: transaction throughput and transaction latency. In the proposed CMS system, the transaction throughput is computed based on the number of peers (M), the number of transactions contained in each peer block *(BS)*, and the time for creating the Super-Block. Instead in traditional Bitcoin such as SMS and PMS, the transaction throughput is computed based on the number of transactions contained in just one miner block (winner block) and the time for the creation of that block.

In the proposed CMS system, the expected time for transferring transaction number "i" from the waiting pool (Memory Pool) to the blockchain is computed based on the actual number of transactions added to the blockchain per second (transaction throughput or T_{coop}) and it is order (i) in the waiting pool. Instead in traditional Bitcoin such as SMS and PMS, the expected time for transferring transaction number "i" from the waiting pool (Memory Pool) to the blockchain is computed based on the actual number of transactions added to the blockchain per second (transaction throughput or T_{comp}) and it is order (i) in the waiting pool. This time is called T_{Final}. So, the latency is the difference between T_{Final} *and* $T_{Arrival}$.

As a proposition, equations (1) and (2) represent the throughput per second and transaction latency of the proposed cooperative CMS system, respectively. Equations (3) and (4) are used to present the same two aspects of the traditional competitive Bitcoin system. The used abbreviations are presented in table 3:

$$T_{coop} = (BS \times M)/SBCT \tag{1}$$

$$L_{i,coop} = \left\lceil \frac{TX_i}{T_{coop}} \right\rceil - T_{Arrival} \tag{2}$$

$$T_{comp} = BS/BCT \tag{3}$$

$$L_{i/comp} = \left\lceil \frac{TX_i}{T_{comp}} \right\rceil - T_{Arrival} \tag{4}$$

Note that, $SBCT$ is equal to BCT plus the time for integration which is a very small value ε. In the traditional Bitcoin, block creation time is 10 minutes. After those 10 minutes, only one block created by only one peer (the firstly created block) is added to the blockchain, and all other blocks created by other peers are ignored. Instead in CMS, the block creation time remains the same 10 minutes, but all the created blocks by all different peers are not ignored and integrated to create the Super-Block which is added to the blockchain. Since $SBCT = BCT + \varepsilon$ and ε is a very small value, then it could be ignored (the time to integrate all the blocks).

TABLE 3. Abbreviations used in the equations and proofs.

Abbreviation	Description
M	Number of miners (peers) in the P2P network
BS	Block Size (number of transactions in the block)
BCT	Block Creation Time in seconds
$SBCT$	Super Block Creation Time in seconds
TX_i	Transaction number i in the MemPool (Transaction order)
T_{coop}	Transaction Throughput per second in the proposed CMS system
T_{comp}	Transaction Throughput per second in the traditional competitive Bitcoin
$L_{i/coop}$	TX_i Latency in the proposed CMS system
$L_{i/comp}$	TX_i Latency in the traditional competitive Bitcoin
$T_{Arrival}$	The time that the transaction arrives to the MemPool
T_{Final}	The time that the transaction is finally stored in the block

The proposition equations (1 - 4) are used to prove the scalability improvement resulting from using the proposed CMS system.

Proof (1): Proof the equation (2) through the following:

$$L_{i/coop} = T_{Final} - T_{Arrival}$$

$$T_{Final} = (\left\lceil \frac{TX_i}{BS \times M} \right\rceil \times SBCT)$$

- *From Equations 1:*

$$L_{i,coop} = \left\lceil \frac{TX_i}{T_{coop}} \right\rceil - T_{Arrival}$$

Proof (2): Proof the equation (4) through the following:

$$L_{i/comp} = T_{Final} - T_{Arrival}$$

$$T_{Final} = (\left\lceil \frac{TX_i}{BS} \right\rceil \times BCT)$$

- *From Equations 3:*

$$L_{i/comp} = \left\lceil \frac{TX_i}{T_{comp}} \right\rceil - T_{Arrival}$$

Proof (3): Proof that T_{coop} is larger than T_{comp} through the following:

- *From Equations 1 and 3:*

$$T_{coop} = (BS \times M)/SBCT$$

- *Note that,*

$$SBCT = BCT + \varepsilon$$

- *Thus,*

$$T_{coop} = (BS \times M)/BCT + \varepsilon$$

- *Where ε is a very small value that could be ignored.*

$$T_{coop} = (BS \times M)/BCT$$

- *So,*

$$\frac{T_{coop}}{T_{comp}} = \frac{(BS \times M)/BCT}{BS/BCT}$$
$$\frac{T_{coop}}{T_{comp}} = \frac{BS \times M}{BCT} \times \frac{BCT}{BS}$$
$$\frac{T_{coop}}{T_{comp}} = M \geq 1$$
$$T_{coop} \geq T_{comp}$$

Proof (4): Proof that $L_{i/coop}$ is smaller than in $L_{i/comp}$ through the following:

- From equations 2 and 4, after adding $T_{Arrival}$ to both equations:

$$L_{i/coop} = \left\lceil \frac{TX_i}{T_{coop}} \right\rceil$$
$$L_{i/comp} = \left\lceil \frac{TX_i}{T_{comp}} \right\rceil$$
$$\frac{L_{i/coop}}{L_{i/comp}} = \frac{TX_i/T_{coop}}{TX_i/T_{comp}}$$
$$\frac{L_{i/coop}}{L_{i/comp}} = \frac{TX_i}{T_{coop}} \times \frac{T_{comp}}{TX_i}$$
$$\frac{L_{i/coop}}{L_{i/comp}} = \frac{T_{comp}}{T_{coop}}$$

- *From proof 1, while:*

$$T_{coop} \geq T_{comp}$$
$$\frac{T_{comp}}{T_{coop}} \leq 1$$

- *So,*

$$\frac{L_{i/coop}}{L_{i/comp}} \leq 1$$
$$L_{i/coop} \leq L_{i/comp}$$

The following examples (1 and 2) (case studies) show the effect of increasing the number of peers and transactions on the throughput and latency. Table 4 and table 5 present the effect of increasing the number of peers on the throughput and the latency in the proposed system versus the traditional Bitcoin system. Table 6 and table 7 present the effect of increasing the number of waiting for transactions on the throughput and the latency in the proposed CMS system versus the traditional Bitcoin system.

Increasing the number of peers Increases the transaction throughput in CMS, where there are no ignored blocks, and all blocks are integrated and added to the blockchain. So, there are M × BS transactions that are added to the blockchain every 10 minutes. Instead, the transaction throughput in the traditional Bitcoin is not affected, whereas there is only one block added to the blockchain. So, there are only BS transactions added to the blockchain every 10 minutes. Also, increasing the number of peers decreases the latency in CMS because the number of transactions that are added to the blockchain has increased, whereas the transaction is waiting less time to be added to the blockchain. In contrast, the latency in traditional Bitcoin is not affected.

The transaction throughput in CMS and traditional Bitcoin is not affected by the number of transactions waiting in the waiting pool. But increasing the number of transactions in the waiting pool causes the transaction to wait longer until it is added to the blockchain (latency increased).

From the previous equations and proofs, we conclude that in the proposed system:

TABLE 4. Example (1) part (1): Computing the transaction throughput when BS = 100 transactions & SBCT = BCT = 10 minutes = 60 seconds.

M = Number of Peers	T_{COOP}	T_{COMP}
5	50 Tx per minute	10 Tx per minute
10	100 Tx per minute	10 Tx per minute
50	500 Tx per minute	10 Tx per minute
100	1000 Tx per minute	10 Tx per minute
500	5000 Tx per minute	10 Tx per minute
1000	10,000 Tx per minute	10 Tx per minute

TABLE 5. Example (1) Part (2): Computing the transaction Latency when BS = 100 transactions & SBCT = BCT = 10 minutes = 60 seconds & number of transactions in the waiting pool = 1000 transactions arrived at the same time ($T_{Arrival}$ = Zero).

M = Number of Peers	$L_{i,COOP}$	$L_{I/COMP}$
5	20 minutes	100 minutes
10	10 minutes	100 minutes
50	2 minutes	100 minutes
100	1 minute	100 minutes
500	1 minute	100 minutes
1000	1 minute	100 minutes

TABLE 6. Example (2) part (1): Computing the transaction throughput when M = Number of peers = 1000 & BS = 100 transactions & SBCT = BCT = 10 minutes = 60 seconds.

Number of Transactions in the waiting pool	T_{COOP}	T_{COMP}
10,000	10,000 Tx per minute	10 Tx per minute
50,000	10,000 Tx per minute	10 Tx per minute
100,000	10,000 Tx per minute	10 Tx per minute
200,000	10,000 Tx per minute	10 Tx per minute

TABLE 7. Example (2) part (2): Computing the transaction Latency when M = Number of peers = 1000 & BS = 100 transactions & SBCT = BCT = 10 minutes = 60 seconds.

Number of Transactions in the waiting pool	$L_{i,COOP}$	$L_{I/COMP}$
10,000	1 minute	1000 minutes
50,000	5 minutes	5000 minutes
100,000	10 minutes	10,000 minutes
200,000	20 minutes	20,000 minutes

1) Throughput is directly proportional to the number of miners (peers) in the network.
2) Transaction latency is inversely proportional to the throughput.

Algorithm 1 presents the proposed algorithm that our Cooperative Mining System depends on.

Algorithm 1 Creating the New Super-Block Algorithm in CMS

```
Input: $TX_s$, Blockchain, Difficulty_Target
Variables:
        Nonce: Integer number starting from zero
        Difficulty_Target: 256 bits in hexadecimal
        Empty_Block_Hash: 256 bits in hexadecimal
        Block_List: Received blocks from other nodes
        Super_Block: The final integrated block
        Transaction_List: List of all transactions
        Final_Block_Hash: 256 bits in hexadecimal
Output: Blockchain includes the final new super-block
1.  New_Block = Create_New_Block ($TX_s$)
2.  Broadcast the New_Block
3.  Empty_Block = Create_Empty_Block ()
4.  Nonce = 0
5.  Empty_Block_Hash = 0
6.  WHILE Empty_Block_Hash<Difficulty_Target DO
        Nonce = Nonce + 1
        Empty_Block_Hash = Calculate_Hash (Nonce)
        IF the node receives blocks from all other nodes
        THEN
            Break
        END IF
    END WHILE
7.  FOR EACH Block B in Block_List DO
        FOR EACH Transaction TX in Block B DO
            Add TX to Transaction_List
        END FOR EACH
        END FOR EACH
8.  Remove the redundant transactions from
    Transaction_List
9.  Sort the Transaction_List according to the hash
    number of each transaction
10. Super_Block = Create_New_Block (Tansaction_List)
11. Add the Super_Block to the blockchain
```

V. IMPLEMENTATION AND EVALUATION RESULTS

The Simulation of the proposed CMS system, Parallel Mining System (PMS) [37], [50] and Bitcoin Solo Mining System (SMS) [27] is developed using C# .NET programming language. These three systems are implemented on the same real peer-to-peer network to compare them. The network is constructed from devices of different capabilities to ensure that miners are different. Also, the server is created to broadcast run commands to all devices ensuring that all devices start mining simultaneously at the beginning.

A. EVALUATION METRICS AND PARAMETERS

The experimentations have been conducted on each of the above three systems depending on three different parameters. These parameters are the number of peers (miners that are competing or cooperating to create the new block), the number of transactions (transactions in *MemPool* that are waiting for confirmation), and difficulty level (degree of mining difficulty) as presented in table 8. In each parameter, we change the value of this parameter and keep the values of the other two parameters unchanged to see the effect of this parameter on the system. The system evaluation criteria are average Block Creation Time (BCT), transaction throughput, and transaction latency.

TABLE 8. Evaluation parameters.

Difficulty	Number of Peers	Number of Transactions in MemPool
3,4,5	2,4,6,8,10,12	40,80,120,160,200,240

B. EXPERIMENTS

1) NUMBER OF PEERS

We set the number of waiting transactions to 100 and the difficulty level to 4. In contrast, we switched the number of peers between 2 to 12 peers to get different case scenarios for the mining process. Table 9, table 10, and table 11 present the resulting values for BCT, transaction throughput, and transaction latency in the three systems.

TABLE 9. Average BCT values in milliseconds for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	120	2	10	9286	8564
		4	18	4190	4426
		6	24	2774	2581
		8	33	2009	2071
		10	45	1714	1630
		12	48	1414	1224

TABLE 10. Transaction throughput values in minutes for CMS, PMS, SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	120	2	12000	6	7
		4	13332	14	13
		6	15000	21	23
		8	14544	29	28
		10	13330	35	36
		12	15000	42	49

We can observe the following from these results:

- In CMS, the average BCT increases when the number of peers increases as shown in Figure 6. The number of blocks that are aggregated to create the super-block

TABLE 11. Last transaction latency values in minutes for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	120	2	0	20	17
		4	0	8	9
		6	0	5	5
		8	0	4	4
		10	0	3	3
		12	0	2	2

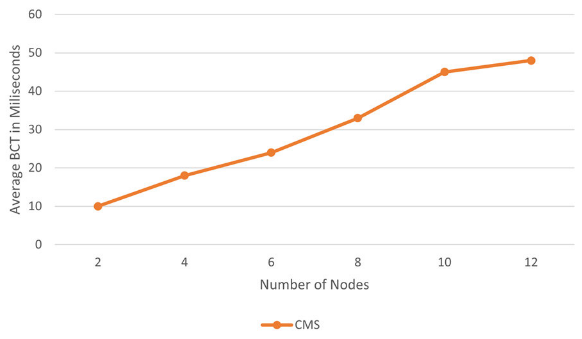

FIGURE 6. Average BCT in milliseconds for the proposed CMS.

increases as the number of peers increases. Therefore, the time required to aggregate these blocks increases as well as the average BCT for the super-block.

- In PMS and SMS, average BCT decreases when the number of peers increases as shown in Figure 7. The hash rate increases as the number of peers increases. So, the block is mined faster.
- Average BCT in CMS is lower than that in PMS and SMS. This can be explained by the fact that PoW is automatically stopped after collecting all blocks from all miners in CMS.
- In the three systems, the transaction throughput increases when the number of peers increases as shown in figure 8 and figure 9. In CMS, all peers create blocks, so the number of transactions will be increased as the number of peers increases. In PMS and SMS, the block is created faster when the number of peers increases, so the number of transactions per second will be increased.
- Transaction throughput in CMS is much more than that in PMS and SMS. Whereas in CMS all peers create blocks but in PMS and SMS only one peer can create a block.
- In the three systems, the transaction latency decreases when the number of peers increases because the transaction throughput increases as shown in figure 10. It is almost non-existent in CMS.

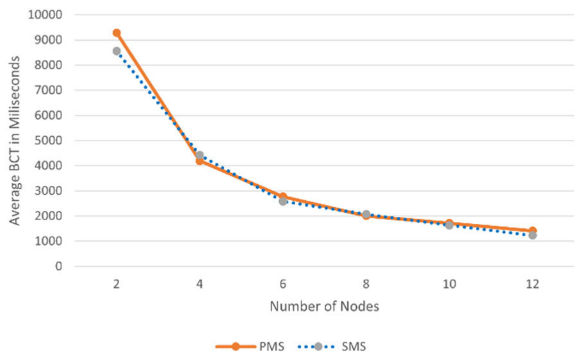

FIGURE 7. Average BCT in milliseconds for both PMS and SMS.

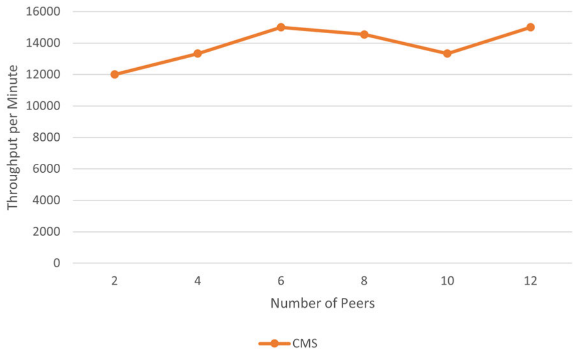

FIGURE 8. Transaction throughput in minutes for the proposed CMS.

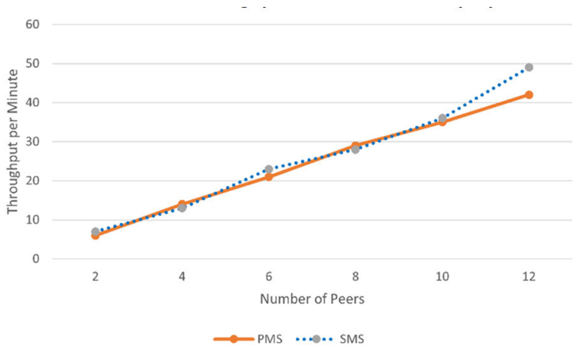

FIGURE 9. Transaction throughput in minutes for both PMS and SMS.

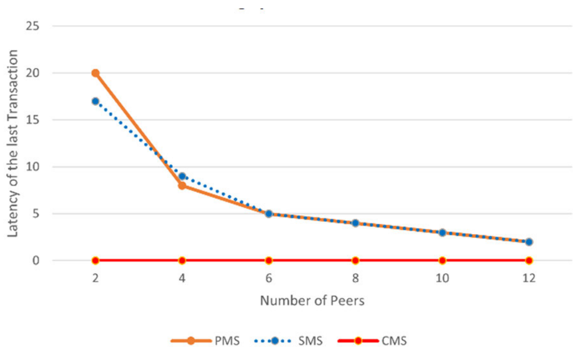

FIGURE 10. Last transaction latency in minutes for CMS, SMS, and PMS.

2) NUMBER OF TRANSACTIONS

In this perspective, we set the difficulty level to 4 and the number of peers to 8 peers, but the number of waiting transactions is switched between 40 to 240 transactions. Table 12, table 13, and table 14 present the resulting values of average BCT, transaction throughput, and transaction latency in the three systems.

TABLE 12. Average BCT values in milliseconds for CMS, PMS, SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	40	8	53	1928	1969
	80		38	1935	2063
	120		42	1942	2130
	160		37	2098	2060
	200		25	2083	2032
	240		27	2012	1960
Average:			**37**	**1977**	**2046**

TABLE 13. Transaction throughput values in minutes for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	40	8	12968	31	30
	80		12624	31	29
	120		11424	30	28
	160		12968	28	29
	200		19200	28	29
	240		17776	29	30
Average:			**12968**	**29**	**29**

TABLE 14. Last transaction latency values in minutes for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
4	40	8	0	1	1
	80		0	2	2
	120		0	4	4
	160		0	5	5
	200		0	7	6
	240		0	8	8

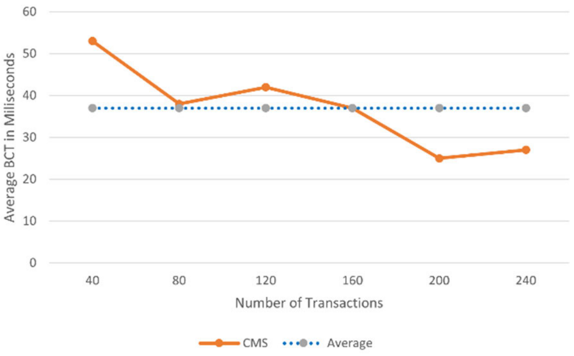

FIGURE 11. Average BCT in milliseconds for the proposed CMS.

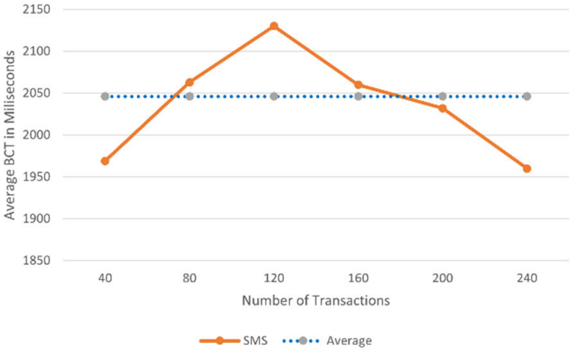

FIGURE 12. Average BCT in milliseconds for SMS.

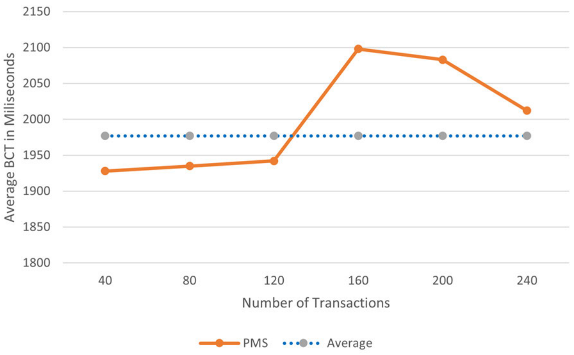

FIGURE 13. Average BCT in milliseconds for PMS.

We can observe the following from these results:

- While changing the number of waiting transactions in *MemPool*, the resulting values of average BCT and transaction throughput are almost unchanged. This indicates that in the three systems, average BCT and transaction throughput are not affected by the number of transactions in the *MemPool* as presented in figures11. . .16.
- On the other hand, in PMS and SMS, transaction latency increases as the number of waiting transactions increases. It is almost non-existent in CMS due to the high transaction throughput of CMS. Figure 17 shows the increase in the last transaction latency in the three systems.

3) THE DIFFICULTY LEVEL

In this case, difficulty levels 3,4, and 5 are used to evaluate the three systems as well as constant values for the number of peers and the number of transactions equal to 8 peers and 40 transactions, respectively. Table 15, table 16, and table 17 present the resulting values of average BCT, transaction throughput, and transaction latency in the three systems.

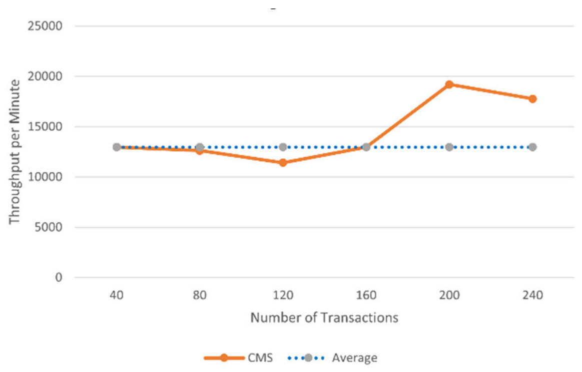

FIGURE 14. Transaction throughput in minutes for the proposed CMS.

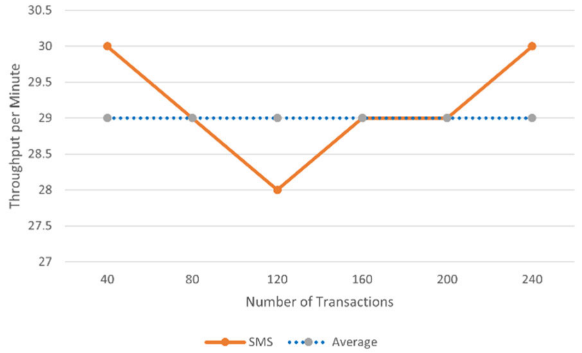

FIGURE 15. Transaction throughput in minutes for SMS.

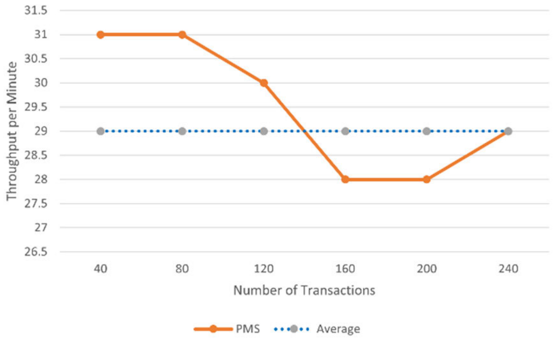

FIGURE 16. Transaction throughput in minutes for PMS.

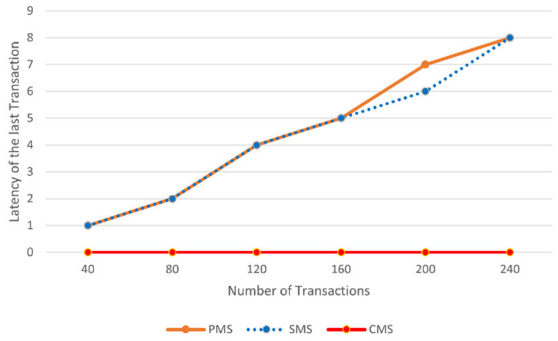

FIGURE 17. Last transaction latency in minutes for CMS, SMS, and PMS.

TABLE 15. Average BCT values in milliseconds for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
3	40	8	59	140	115
4			59	2486	2215
5			49	43199	35742

TABLE 16. Transaction throughput values in minutes for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
3	40	8	8128	428	521
4			8128	24	27
5			9792	1	1

TABLE 17. Last transaction latency values in minutes for CMS, PMS, and SMS.

Difficulty	Number of Transactions in MemPool	Number of Peers	CMS	PMS	SMS
3	40	8	0	0	0
4			0	1	1
5			0	40	40

We can observe the following from these results:

- Average BCT increases when the difficulty level increases in both SMS and PMS, but it is almost constant in CMS as displayed in figure 18 and figure 19. The PoW is automatically stopped when all miners receive all blocks from other miners. Therefore, the Average BCT in CMS is almost constant and lower than that in PMS and SMS.
- Transaction throughput decreases as the difficulty level increases in both PMS and SMS due to the increase in average BCT. In contrast, in CMS it is almost as constant as the average BCT. Figure 20 and figure 21 show the transaction throughput in the three systems.
- In PMS and SMS, transaction latency increases as the difficulty level increases due to the decrease in the transaction throughput. In CMS, it is almost non-existent as shown in figure 22.

C. RESULTS DISCUSSION

The increase in the number of peers:

- Increases the transaction throughput in CMS. Whereas there are no ignored blocks, and all blocks are integrated and added to the blockchain. So, there are M × BS transactions that are added to the blockchain every epoch.
- Does not affect the transaction throughput in PMS and SMS. Whereas there is only one block added to the

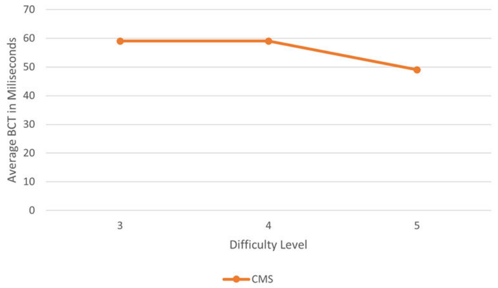

FIGURE 18. Average BCT in milliseconds for the proposed CMS.

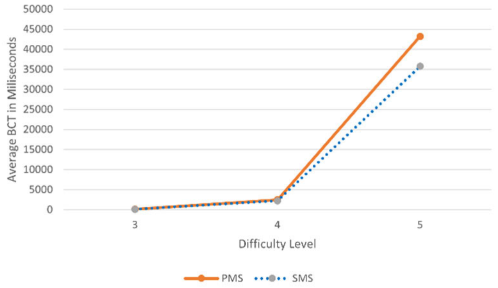

FIGURE 19. Average BCT in milliseconds for both PMS and SMS.

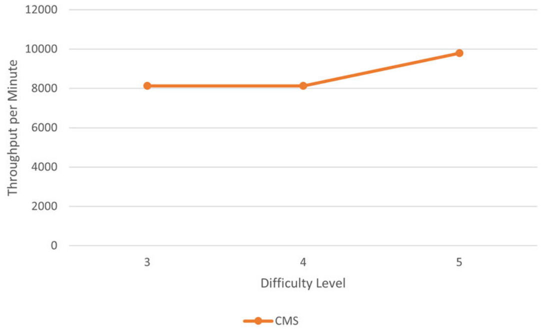

FIGURE 20. Transaction throughput in minutes for the proposed CMS.

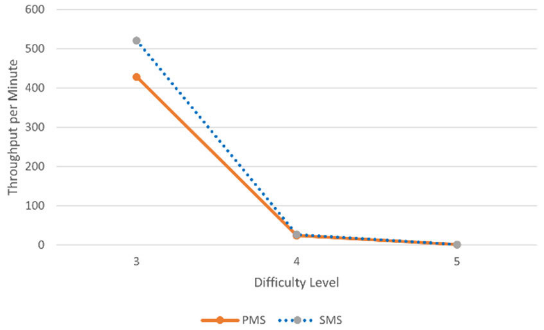

FIGURE 21. Transaction throughput in minutes for both PMS and SMS.

blockchain. So, there are only BS transactions added to the blockchain every epoch.

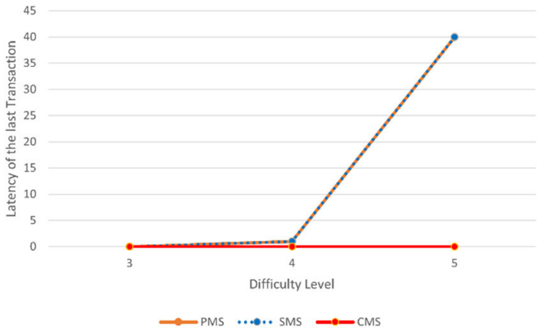

FIGURE 22. Last transaction latency in minutes for CMS, SMS, and PMS.

- Decreases the latency in CMS because the number of transactions that are added to the blockchain is increased. Whereas the transaction waits less time to be added to the blockchain.
- In PMS and SMS the latency is not affected as the transaction throughput is not affected.

Transaction throughput in CMS, PMS, and SMS is not affected by the number of transactions waiting in *MemPool*. But increasing the number of transactions in *MemPool* causes the transaction to wait longer until it is added to the blockchain (latency increased). That is for all systems CMS, PMS, and SMS, but in CMS, it is still a small time.

From the previous results we can conclude that:

1) The proposed CMS significantly increases transaction throughput compared to both PMS and SMS. In CMS, all peers create blocks but in PMS and SMS, only one peer can create a block.
2) Transaction latency is almost eliminated in the proposed CMS due to its high transaction throughput. In contrast, PMS and SMS transactions suffer from confirmation delays.

VI. CONCLUSION AND FUTURE WORK

In this paper, we focus on solving the scalability problem of Bitcoin without affecting decentralization and security level which is called blockchain trilemma. This paper focuses on improving Bitcoin scalability in terms of transaction throughput and latency. We propose a CMS system where miners cooperate but do not rely on each other to mine the new block rather than the competition between miners in the Bitcoin system. Therefore, the rewards are divided equally among miners. The proposed CMS significantly increases the transaction throughput compared to the traditional Bitcoin system. It also nearly eliminates transaction delays and starvation. We have achieved this improvement by modifying the proof of work algorithm of Bitcoin which has been proven mathematically.

In the future, we plan to implement the proposed system in the cloud such as Amazon EC2 so that we can increase the size of the network. Also, as a result of increasing transaction throughput, the size of the blockchain is growing rapidly and it is difficult to store it in miners' computers. So, we can solve this problem by changing the data structure of the blockchain to allow it to store only the last state of the users instead of storing all transactions. Finally, a scheduler that chooses transactions randomly can be developed in the future.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, no. 2022, p. 21260, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[2] *Bitcoin*. Accessed: 2022. [Online]. Available: https://bitcoin.org/en/
[3] *Ethereum*. Accessed: 2022. [Online]. Available: https://ethereum.org/en/
[4] *Iota*. Accessed: 2022. [Online]. Available: https://www.iota.org/
[5] D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," Ripple Labs Inc, White Paper, San Francisco, CA, USA, 2014, vol. 5, no. 8, p. 151.
[6] *Bitcoin Cash*. Accessed: 2022. [Online]. Available: https://www.bitcoincash.org/
[7] *Cardano*. Accessed: 2022. [Online]. Available: https://cardano.org/
[8] *Litecoin*. Accessed: 2022. [Online]. Available: https://litecoin.org/
[9] *Monero*. Accessed: 2022. [Online]. Available: https://www.getmonero.org/
[10] *Neo*. Accessed: 2022. [Online]. Available: https://neo.org/
[11] E. Duffield and D. Diaz, "Dash: A payments-focused cryptocurrency," White Paper, 2018. [Online]. Available: https://github.com/dashpay/dash/wiki/Whitepaper
[12] *CoinMarketCap*. Accessed: 2022. [Online]. Available: https://coinmarketcap.com/
[13] M. S. Ferdous, M. J. M. Chowdhury, and M. A. Hoque, "A survey of consensus algorithms in public blockchain systems for crypto-currencies," *J. Netw. Comput. Appl.*, vol. 182, May 2021, Art. no. 103035.
[14] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125244–125262, 2020.
[15] S. P. Yadav, K. K. Agrawal, B. S. Bhati, F. Al-Turjman, and L. Mostarda, "Blockchain-based cryptocurrency regulation: An overview," *Comput. Econ.*, vol. 57, pp. 1–17, Oct. 2020.
[16] P. Müller, S. Bergsträßer, A. Rizk, and R. Steinmetz, "The Bitcoin universe: An architectural overview of the Bitcoin blockchain," in *Proc. 11th DFN-Forum Kommunikationstechnologien, Gesellschaft für Informatik eV*, 2018, pp. 1–20.
[17] S. Goswami, "Scalability analysis of blockchains through blockchain simulation," Univ. Nevada, Las Vegas, Las Vegas, NV, USA, 2017.
[18] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.
[19] V. G. Martínez, L. Hernández-Álvarez, and L. H. Encinas, "Analysis of the cryptographic tools for blockchain and Bitcoin," *Mathematics*, vol. 8, no. 1, p. 131, Jan. 2020.
[20] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[21] A. Abidi, B. Bouallegue, and F. Kahri, "Implementation of elliptic curve digital signature algorithm (ECDSA)," in *Proc. Global Summit Comput. Inf. Technol. (GSCIT)*, Jun. 2014, pp. 1–6.
[22] N. M. Hamza, S. Ouf, and I. M. El-Henawy, "Exploring blockchain mining mechanism limitations," in *Proc. Int. Conf. Innov. Comput. Commun.* Singapore: Springer, 2021, pp. 749–757.
[23] N. T. Courtois, M. Grajek, and R. Naik, "Optimizing SHA256 in Bitcoin mining," in *Proc. Int. Conf. Cryptogr. Secur. Syst.* Berlin, Germany: Springer, 2014, pp. 131–144.
[24] E. H. Umucu, "Elliptic curve cryptography in blockchain technology," *SSRN*, 2022, Art. no. 4033934, doi: 10.2139/ssrn.4033934.
[25] H. Hellani, A. E. Samhat, M. Chamoun, H. E. Ghor, and A. Serhrouchni, "On BlockChain technology: Overview of Bitcoin and future insights," in *Proc. IEEE Int. Multidisciplinary Conf. Eng. Technol. (IMCET)*, Nov. 2018, pp. 1–8.
[26] A. I. Badev and M. Chen, "Bitcoin: Technical background and data analysis," *SSRN*, 2014, Art. no. 2544331.
[27] J. Frankenfield. *Proof of Work (PoW)*. Accessed: 2022. [Online]. Available: https://www.investopedia.com/terms/p/proof-work.asp
[28] A. Barhanpure, P. Belandor, and B. Das, "Proof of stack consensus for blockchain networks," in *Proc. Int. Symp. Secur. Comput. Commun.* Singapore: Springer, 2018, pp. 104–116.
[29] S. M. S. Saad, R. Z. R. M. Radzi, and S. H. Othman, "Comparative analysis of the blockchain consensus algorithm between proof of stake and delegated proof of stake," in *Proc. Int. Conf. Data Sci. Its Appl. (ICoDSA)*, Oct. 2021, pp. 175–180.
[30] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OsDI*, vol. 99, 1999, pp. 173–186.
[31] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021.
[32] 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. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 3–16.
[33] S. Jiang and J. Wu, "Taming propagation delay and fork rate in Bitcoin mining network," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Dec. 2021, pp. 314–320.
[34] D. Fullmer and A. S. Morse, "Analysis of difficulty control in bitcoin and proof-of-work blockchains," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5988–5992.
[35] C. Bertucci, L. Bertucci, J.-M. Lasry, and P.-L. Lions, "Economic modelling of the Bitcoin mining industry," *SSRN*, 2021, Art. no. 3907822.
[36] S. Dexter. *Longest Chain—How Are Blockchain Forks Resolved?* Accessed: 2022. [Online]. Available: https://www.mangoresearch.co/blockchain-forks-explained/
[37] S. S. Hazari and Q. H. Mahmoud, "Improving transaction speed and scalability of blockchain systems via parallel proof of work," *Future Internet*, vol. 12, no. 8, p. 125, Jul. 2020.
[38] M. Barrett. *An Introduction to Directed Acyclic Graphs*. Accessed: 2022. [Online]. Available: https://cran.r-project.org/web/packages/ggdag/vignettes/intro-to-dags.html
[39] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling Nakamoto consensus to thousands of transactions per second," 2018, *arXiv:1805.03870*.
[40] S. Popov, "The tangle," White Paper, 2018, vol. 1, no. 2022. [Online]. Available: https://iota.org/IOTA_Whitepaper.pdf
[41] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2016, pp. 45–59.
[42] M. Kedziora, D. Pieprzka, I. Jozwiak, Y. Liu, and H. Song, "Analysis of segregated witness implementation for increasing efficiency and security of the bitcoin cryptocurrency," *J. Inf. Telecommun.*, vol. 7, no. 1, pp. 1–12, 2022.
[43] *Bitcoin Classic*. Accessed: 2022. [Online]. Available: https://bitcoinclassic.com/
[44] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. Int. Conf. Manage. Data*, Jun. 2019, pp. 123–140.
[45] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 17–30.
[46] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," 2017, *arXiv:1708.03778*.
[47] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 931–948.
[48] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 95–112.
[49] Z. H. Chin, T. T. V. Yap, and I. K. T. Tan, "Simulating difficulty adjustment in blockchain with SimBlock," in *Proc. 2nd ACM Int. Symp. Blockchain Secure Crit. Infrastruct.*, Oct. 2020, pp. 192–197.
[50] S. S. Hazari and Q. H. Mahmoud, "A parallel proof of work to improve transaction speed and scalability in blockchain systems," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0916–0921.

DALIA ELWI received the B.Sc. degree in computer science from the Faculty of Computers and Information, Mansoura University, Egypt, in 2013, and the M.Sc. degree in computer science, in 2018.

She is currently an Assistant Lecturer with the Department of Computer Science, Faculty of Computers and Information, Mansoura University. Her current research interests include artificial intelligence, cryptocurrency, bitcoin, blockchain, distributed systems, the Internet of Things, and software engineering.

OSAMA ABU-ELNASR received the M.S. degree in computer vision and the Ph.D. degree in empowering software testing processes for improving software quality from Mansoura University, in 2009 and 2014, respectively.

He is currently an Associate Professor with the Computer Science Department, Faculty of Computers and Information, Mansoura University. He participated in the reviewing process of many indexed refereed computer science journals. His research interests include information security, the IoT, artificial intelligence, high-performance computing, and software quality management.

AHMED S. TOLBA received the B.Sc. and M.Sc. degrees from the College of Engineering, Mansoura University, Egypt, in 1978 and 1981, respectively, and the Dr.Ing. degree in electrical and computer engineering from Wuppertal University, Germany, in 1988.

He was a Professor of computer engineering with the College of Engineering, Suez Canal University, from 1989 to 2000. He was also with the Applied Physics Department, Kuwait University, from 1995 to 2002. He was the Director of the E-Learning Center, Mansoura University, from 2003 to 2007. He has renewed the validation of the AOU's computer studies problem from The Open University, U.K. He was the Director of the Information and Communication Technologies Project (ICTP), Ministry of Higher Education in Egypt, from 2003 to 2004, the Director of the National E-Learning Center, the Director of knowledge and electronic services (eKSc) with the Supreme Council of Egyptian Universities, from 2006 to 2007 and 2014, and the Executive Director of the Project Management Unit (PMU), Ministry of Higher Education, from 2014 to 2016. He has been the xDean of the College of Computer and Information Sciences, Mansoura University, from 2004 to 2007, and Arab Open University (AOU), Kuwait, from 2007 to 2013. He was the Vice President for Education and Information Technologies with AOU in eight countries in its headquarters in Kuwait, from 2012 to 2013, and MISR University for Science and Technology, Egypt, from 2018 to 2019. Since 2021, he has been the Dean of the Institute of Engineering and Automotive and Renewable Energy, New Heliopolis Engineering Institute, Cairo. He is currently a Professor of computer science with the College of Computer and Information Sciences, Mansoura University. He has authored or coauthored around 65 publications, including refereed IEEE/Sage/Springer/Elsevier journals, conference papers, and book chapters. His current research interests include the Internet of Things, fog computing, quantified self, smart health, pattern recognition, gesture recognition, computer vision, image watermarking, medical image analysis, machine vision, activity recognition, cognitive computing, blockchain, smart homes, and digital twins.

SAMIR ELMOUGY received the Ph.D. degree in computer science from the School of Electrical Engineering and Computer Science, Oregon State University, USA.

From 2008 to 2014, he was an Assistant Professor with the Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is currently a Professor of computer science and the Vice Dean for Postgraduate Studies and Research with the Faculty of Computers and Information, Mansoura University, Egypt. He participated in the reviewing process of many international refereed journals and conferences. He has published over 80 articles in refereed journals, such as IEEE Transactions/Springer journals, IEEE conferences, and book chapters. His current research interests include artificial intelligence, the IoT, information theory, and software engineering.