An Efficient, Secured, and Infinitely Scalable Consensus Mechanism for Peer-to-Peer Energy Trading Blockchain

Yingsen Wang, *Student Member, IEEE*, Yixiao Li, Weihan Jiao, Guibin Wang, Juanjuan Zhao, Yan Qiang, and Keqin Li, *Fellow, IEEE*

***Abstract*—Peer-to-Peer (P2P) energy trading has the potential to create a more efficient and sustainable electricity market. It can relieve the load on the power grid during peak usage periods. However, such a P2P energy market is decentralized and vulnerable to numerous cyber attacks. Decentralized blockchain technology has been proposed to solve these problems. The consensus mechanism is the core of blockchain. It determines the effectiveness and safety of blockchain for energy trading. However, consensus mechanisms applied to energy trading today are traditional consensus. Due to their high latency and significant computational power, they cannot be directly implemented in P2P energy trading. Therefore, we propose a novel Block Alliance Consensus (BAC) mechanism. The BAC breaks through the blockchain impossible triangle in the energy trading scenario by reasonably pursuing decentralization. We achieve infinite scalability via sharding. Within each shard, we substitute the Hashgraph for conventional methods to further improve the throughput and transaction speed. We design a cross-shard method for transactions between different shards. We implement the energy trading blockchain (ETB) and BAC consensus mechanism on the Hyperledger Fabric platform. The experiments show that our ETB is not limited to the impossibility triangle like other consensus. Our BAC mechanism achieves infinite scalability while ensuring high levels of security.**

Manuscript received 22 January 2023; revised 29 March 2023; accepted 30 April 2023. Date of publication 12 May 2023; date of current version 20 September 2023. Paper 2023-ESC-0105.R1, presented at the 2022 IEEE Industry Applications Society Annual Meeting, Detroit, MI, USA, Oct. 09–14, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Energy Systems Committee of the IEEE Industry Applications Society [DOI: 10.1109/IAS54023.2022.9939776]. This work was supported in part by State Grid Corporation Headquarters Technology under Grant 5400-202216167A-1-1-ZN and in part by the National Natural Science Foundation of China under Grants U21A20469, 61872261, and 61972274. *(Corresponding author: Yan Qiang.)*

Yingsen Wang, Juanjuan Zhao, and Yan Qiang are with the College of Information and Computer Science, Taiyuan University of Technology, Taiyuan 030024, China (e-mail: wangyingsen0065@link.tyut.edu.cn; zhaojuanjuan@tyut.edu.cn; qiangyan@tyut.edu.cn).

Yixiao Li is with the School of Electrical Engineering, Shandong University, Jinan 250300, China (e-mail: 202034669@mail.sdu.edu.cn).

Weihan Jiao is with the School of Electric and Electronic Engineering, North China Electric Power University, Beijing 102206, China (e-mail: 120212201610@ncepu.edu.cn).

Guibin Wang is with the Shandong Power Company of the State Grid, Shandong University, Jinan 250300, China (e-mail: wgb369@sohu.com).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TIA.2023.3275551.

Digital Object Identifier 10.1109/TIA.2023.3275551

***Index Terms*—Blockchain, consensus, energy trading, Hashgraph, peer-to-peer (P2P).**

I. INTRODUCTION

IN THE conventional energy trading market, all energy management operations are carried out by a centralized third-party control center [1], [2], which also makes decisions on generation, transmission, distribution, and delivery [3]. In recent years, existing energy trading systems have started to look beyond their capacities due to the surge in electricity and the promotion of new energy power generation [4]. In contrast, individuals in a P2P energy trading market can be both producers and consumers (prosumers) who generate and sell extra electricity, or just consumers without their own power facilities. Also, smart grids are becoming more common, using communication technology and interconnections to increase the use of renewable energy and tackle energy problems. [5], [6].

P2P energy trading enables individuals to participate in energy management operations, with the potential to increase energy efficiency and promote the use of renewable energy resources [7]. A decentralized digital platform maintained by government or corporate agencies and market participants is needed to ensure secure and transparent transactions.

Decentralized energy trading may be able to overcome the concerns with the conventional structure, but it also brings new challenges, such as security, privacy, and trust issues, which call for new technologies [8], [9]. Using of blockchain technology in energy trading is a promising solution to address security, privacy, and trust issues in decentralized energy trading [10], [11]. Blockchain is a decentralized, distributed, and immutable ledger that consists of an irrevocable sequence of blocks [12], [13]. Blockchain enables the creation and maintenance of a distributed and immutable ledger without the need for a central trusted authority, reducing the likelihood of a single point of failure [14], [15]. Attackers in blockchain must possess a majority of the network's mining power to conduct a successful attack [16], [17]. Although the blockchain originated from digital currencies [18], [19], it can be applied to many other non-monetary scenarios. Blockchain is attracting enormous attention to energy trading and promoting trusted smart grid developments toward decentralization.

The consensus mechanism is the core technology but the bottleneck of the blockchain. It is a crucial blockchain component that allows nodes to agree on new blocks added to the chain. It ensures the validity of data and makes the network fault-tolerant [20], [21]. The Byzantine problem is always the most challenging in the distributed consensus protocol. Popular Byzantine Fault Tolerance (BFT) based consensus protocols include Proof-of-Work (PoW), Proof-of-Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT). PoW and PoS are commonly used consensus mechanisms for digital currencies but unsuitable for the Internet of Things (IoT), such as energy trading. On the other hand, numerous studies have adopted PBFT or upgraded versions of PBFT (such as Ref. [22]) as the consensus mechanism in energy trading. However, the communication complexity of PBFT is too high to be directly used in energy trading. In a word, the majority of current blockchain consensus methods demand a lot of computing power and have high latency, making them unsuitable for real-world P2P energy trading where resources are limited [23].

It is worth mentioning that many studies take the consensus mechanism as a small part of the research and directly adopt the above traditional blockchain consensus mechanisms. The current consensus methods of blockchain have limitations in terms of throughput and scalability. We have developed a new approach using sharding, which allows for infinite scalability (scale-out, i.e., unbounded throughput). In the context of energy trading, we break through blockchain's impossible triangle (security, decentralization, and scalability). We prioritize security and unlimited scalability over decentralization. While we sacrifice some decentralization, we believe it is reasonable to maintain centralization to enable government regulation of the energy economy. The significance of decentralization is that it can ensure security and improve efficiency.

Our research suggests a consortium blockchain for energy trading with a BAC algorithm to ensure data accuracy, reliability, and network security. BAC allows for flexible node management while maintaining Hashgraph's efficiency and security, and it can resist Sybil Attacks in large-scale networks. The preliminary work for this article was presented at the 2022 IEEE Industry Applications Society Annual Meeting [1]. The main contributions of our study are as follows:

1) A distributed Energy Trading Blockchain (ETB) structure is proposed; all the participants record and maintain the ledger together. The ETB has three types of nodes: Primary Node (P), Candidate Primary Node (CP), and Consensus Node (CS), each serving a specific role in the energy trading scenario.
2) We propose the BAC algorithm ideal for the ETB using the revolutionary Hashgraph. The BAC reduces the time complexity of traditional BFT to $O(N)$, where N is the number of nodes. BAC can improve the throughput and security of Hashgraph-based ETB and handle node addition and deletion while preventing Sybil Attacks. BAC is suitable for industrial ETB with limited resources because it does not rely heavily on computational power. The election protocol and incentives ensure safety and progress in consensus despite failures. Experiments show that this platform can handle real-time power transactions and reduce grid load during peak consumption.
3) We use the idea of sharding to reach infinite scaling, and by keeping the centralized component reasonable, we break the impossible triangle of blockchain in energy trading. In addition, we design an efficient cross-shard approach for transactions between shards. We show the advantages of our methods over other conventional as well as some advanced models in terms of security and throughput through experiments and simulations.

The remainder of the article is organized as follows. Section II presents background and related work. Section III introduces our ETB model and the structure of nodes in energy trading platform. The BAC consensus mechanism using Hashgraph is proposed in Section IV. Section V provides the performance of the energy trading platform using our BAC consensus mechanism. The last section concludes our work and outlines future research.

II. BACKGROUND AND RELATED WORKS

A. Blockchain and Distributed Energy Trading

Due to chaining with hash functions cryptographic signature for linking the blocks, it can be ensured that signed blocks cannot tamper with [24]. Numerous important studies on P2P, Blockchain, and consensus-based approaches have been published. Gough et al. [25] discussed the use of an unique multilevel transactive energy optimization model based on a blockchain to schedule distributed energy resources (DERs) inside networked virtual power plants. Gough et al. [26] proposed new rules governing prosumer self-consumption paired with blockchain and new legislation to create an automated energy trading system for residential end-users in local energy markets. Santos-Gonzalez et al. [27] developed a stochastic optimization model for optimal operation of the active distribution networks.

B. Byzantine Fault Tolerance and Blockchain Consensus Mechanism

Lamport et al. [28] proposed the Byzantine Generals Problem (BGP). The BGP is a common challenge that decentralized computer systems must overcome. Its goal is to develop an algorithm that ensures trustworthy generals can come to an agreement. However, the asynchronous BGP is not taken into account. The time threshold t in the traditional BFT (BGP) is a fixed constant value. Asynchronous systems, however, have no sense of the time threshold. The strategy used by PBFT [29] is as follows: the t in PBFT proposed after the traditional BFT will increase if the system times out. It guarantees that nodes will eventually establish a consensus regardless of how much the system's delay is, provided that the delay does not continue to grow indefinitely.

Since theoretical issues that PBFT cannot resolve are unlikely to arise in actual use, the development of BFT appears to have come to a halt. Then, Zyzzyva's suggestion encouraged the continued advancement of BFT [30]. The author believed that the algorithm might be the final solution to the BFT problem. The time complexity of PBFT is $O(N^2)$. Zyzzyva is a speculative-based algorithm: if the primary node is reliable, it is sufficient to conduct a round of routine broadcasts with $O(N)$ message complexity without the need for such a complicated algorithm. The system switches back to the PBFT algorithm if the node determines that the primary is flawed (malicious or down).

C. Energy Trading and Consensus Mechanism

Studies that are pertinent to the use of the BFT-based consensus algorithm in energy trading have been conducted. Sheikh et al. [31] focused on the Byzantine consensus-based distribution network's energy transaction procedure between electric vehicles. Feng et al. [32] presented a scalable, dynamic multi-agent hierarchical PBFT algorithm (SDMA-PBFT), which reduces the communication overhead from $O(N^2)$ to $O(nk\log_k n)$. Wang et al. [33] suggested a voting incentive and penalty system, credit evaluation scheme, and PBFT-based consistency protocol. Wang et al. [34] suggested an algorithm for multi-energy interacting entities based on the PBFT. Cai et al. [35] provided a DPBFT appropriate for the energy blockchain's dynamic reputation.

Bansal and Bhatia [36] proposed an energy trading architecture among electric vehicles (EVs) by directly using Hashgraph. However, the geographic location of EV nodes is changing in real-time. Dynamic addition and removal of nodes are not allowed in Hashgraph. García-Hernández et al. [37] presented an energy trading mechanism in microgrids based on IOTA Tangle. However, the Tangle consensus requires a centralized tool to ensure the network's security. Tangle does not have a global consensus mechanism like blockchain, so security and reliability concerns exist. Abishu et al. [22] suggested an improved PBFT consensus implemented in the Vehicle-to-Vehicle energy trading. However, the time complexity of its improved version is the same as original PBFT. Wang et al. [38] proposed a power trading model within the microgrids based on DAG. However, nodes in DAG constantly process and validate each transaction, which results in higher requirements for transaction processing speed and cost, limiting its applicability in scale.

Although the aforementioned research addresses the issue of low node involvement, it leaves open the issues of high transaction latency and low throughput. Most studies employ the consensus method as a tiny component of the research, and most adopt the standard or an upgraded PBFT as their consensus mechanism. They are unable to satisfy the requirements of massive energy transfers. Therefore, it is vitally necessary to improve the performance of the consensus process on the current energy trading platform.

III. PROPOSED BLOCKCHAIN FRAMEWORK FOR ENERGY TRADING

In the conference version of our paper, we proposed the idea of combining Hashgraph with our previous BAC consensus algorithm, and verified its feasibility through simulation. In this article, we further extend and analyze the innovative consensus mechanism, introducing new techniques such as sharding and cross-shard mechanisms. In the experimental section, we not only compare our approach with some classic consensus mechanisms, but also evaluate its performance against several novel consensus algorithms developed in recent years using multiple performance metrics. We have added several sets of experiments to demonstrate the superiority of our method. We also provide a detailed account of the testing system deployment process and the tested algorithms.

Since Satoshi Nakamoto released the Bitcoin white paper [39], blockchain technology has gained popularity. Blockchain is a distributed ledger system that includes consensus methods, P2P transmission, distributed storage, smart contracts, and distributed ledger technology. The data layer, network layer, consensus layer, incentive layer, contract layer, and application layer make up the blockchain architecture, as depicted in Fig. 1. The detailed benefits of P2P energy trading could be found in Appendix B (Supplementary material).

The virtual trading mechanism between households and electrical grids is shown in Fig. 2 along with transaction data. The energy exchange process between customers and power grids is initiated to solve the issue of demand-supply mismatch. Users receive electricity from the power grid (e.g., smart homes, commercial and industrial consumers). The operation of the power grid is unpredictable due to the intermittent nature of consumer energy demand. The additional load can be supplied by consumers if the electrical grid becomes overloaded during times of high power consumption. Consensus is the method of data transmission and verification that serves as the blockchain's "soul," according to the "General Procedure of Blockchain." We move the detailed descriptions of P2P energy trading's overall model to Appendix E (Supplementary material).

Energy nodes determine their roles based on their energy status and demands, with buyers and sellers communicating through government departments or regulatory authorities. The optimal trading solution is determined using the pricing mechanism outlined in Appendix A (Supplementary material). Upon completion of the transaction, the primary node broadcasts the transaction information within the shard, marking the beginning of the core stage of our ETB - the consensus mechanism discussed in this article.

The BAC consensus is presented within the consortium blockchain's structure. The precise arrangement of nodes is shown in "General Procedure of Blockchain" in Fig. 3. Cli is the name of the client that initiates transaction requests. CA serves as the standard certificate authority. Each network node's identification certificate is managed by CA, and it also creates digital certificates, registers entity IDs on the blockchain, and manages certificate renewals and revocations. The consortium blockchain requires all nodes to be registered and to have a certificate from CA. Each node in the channel maintains the ledger (L), which is the chain and current state information of the channel. Realizing the organization's isolation in the blockchain is the channel's responsibility.

An energy trading blockchain (ETB) node might independently verify the accuracy of a record by using the ETB. Primary Node (P), Candidate Primary Node (CP), and Consensus Node are the three types of ETB nodes (CS). The ETB is implemented on a permissioned, consortium-based blockchain. Nodes without authorisation are not permitted to join, and each node

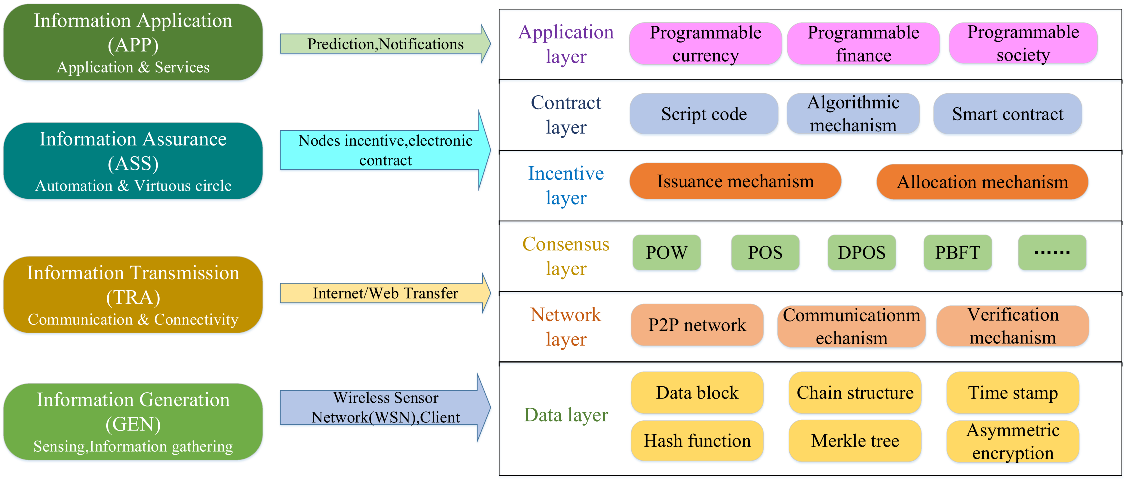


Fig. 1. Composition of the energy trading blockchain.

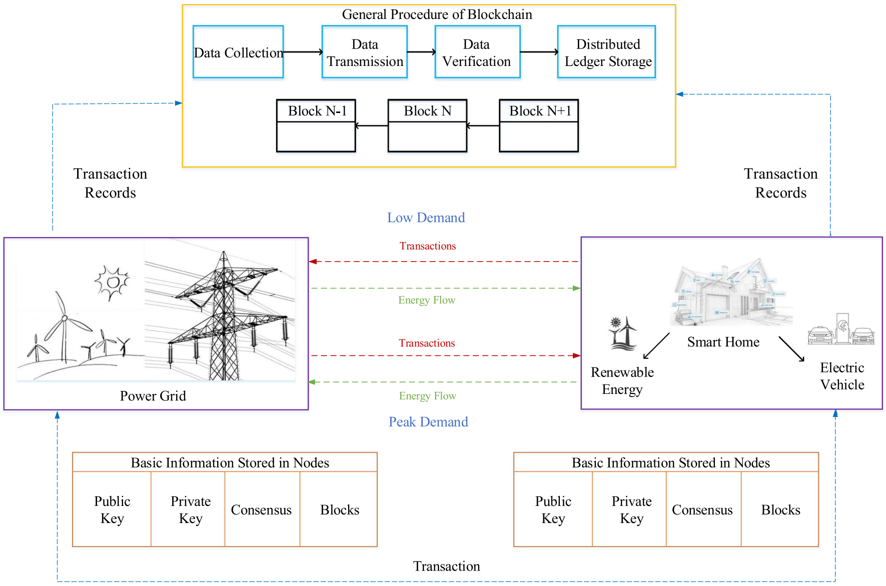


Fig. 2. Blockchain based energy trading process between households and power grids.

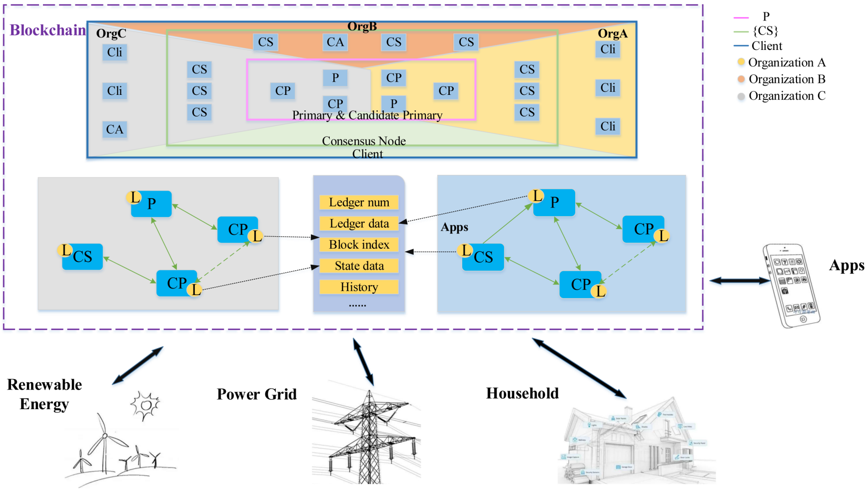


Fig. 3. Detailed structure of nodes.

participating in ETB is subject to authentication. The consortium blockchain's built-in feature initially ensures the security of the ETB. The "Org" in Fig. 3 may be seen as the organization that manages a variety of cooperative businesses. The ETB enables businesses to simultaneously participate in a number of distinct blockchain networks by providing channels that efficiently share infrastructure while protecting data and communication privacy. Our main work focuses on the consensus mechanism between nodes, which is currently the most crucial property in the ETB.

The P can be seen as a power plant in an electric power system. It is responsible for generating new electricity (i.e., transactions), packaging them into blocks, and sending them to other nodes. In an electric power system, power plants are also the primary producers of electricity, generating power and injecting it into the grid for consumption by other users.

CP nodes can be interpreted as grid control centers in an electric power system. They receive and verify transactions from power plants (i.e., primary nodes), package them into blocks, and send them to other nodes. In an electric power system, grid control centers are responsible for coordinating the operation of the grid. They manage the operations of power plants and regulate their output to ensure smooth and safe grid operation.

The CS are power consumers in an electric power system. They store blocks received from the P and CP and validate their integrity. In an electric power system, power consumers are the ultimate electricity users, consuming the electricity transmitted through the grid to satisfy their needs. Additionally, they are essential participants in the electric power system as their demand determines the operation of power plants and the dispatch decisions made by grid control centers.

IV. BAC CONSENSUS MECHANISM

During periods of peak electrical demand, the traditional power grid may experience excessive load which could result in limited electricity supply [40]. The inherent nature of P2P energy trading enables it to alleviate pressure on the power grid during peak usage periods. However, this distributed trading model has yet to be widely implemented in practical life, making the application of P2P energy trading a key issue. The most significant contribution of our research is to move blockchain-based P2P energy trading from the theoretical realm to actual implementation. Through experimentation, we have demonstrated that our method and system not only improve performance but can also be applied on a large scale in real life with the support of legal and policy frameworks.

The BFT-based consensus method describes how blockchain nodes come to a consensus when Byzantine nodes are present (evil or downtime). The choices made by these equal nodes influence many phases of the trade of energy. Similarities between P2P energy trading and BFT can be found in the Byzantine army commander's decision of whether to attack or retreat, which determined whether the operation was successful or unsuccessful.

A. The Original Model of BAC

We first describe the block's structure and the original BAC model to help readers comprehend our BAC mechanism. The BAC model is depicted in Fig. 4. At each level of the BAC, malicious and faulty nodes are both taken into account. The BAC consensus mechanism's specifics are as follows:

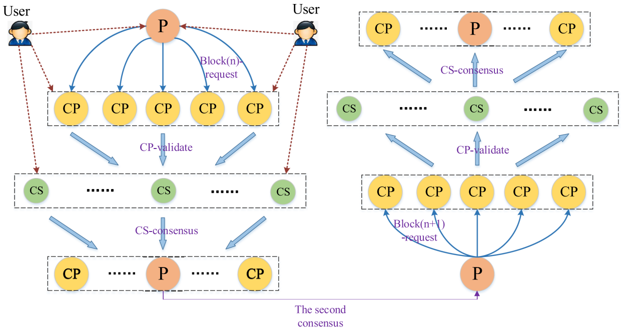


Fig. 4. Original mechanism of BAC.

1) Block-Request: The client broadcasts to all nodes the transaction information. The P transmits the necessary information to the CP by putting it in the current $block(i)$. The request to be linked at height i on the blockchain is made by the unvalidated $block(i) = (header(i), data(i))$. A validated block at height i is represented by the $BLOCK(i)$. The $Prev\_Hash$ stored in $header(i)$ serves as the basis for the verification. The CP determines the hash $H(II)$ of $BLOCK(i-1)$ that it already owns and has already confirmed. The CP delivers its authenticated message $\langle CP-validate, CP(j), H(block(i)), CP(r), CP(c)\rangle_{\sigma_j}$ if $H(II) = Prev\_Hash$.

2) Block-Commit: The ETB completes the first round of consensus in the Block-commit stage. Block authentication messages are acquired by the P and CP from the CS. A CS may misbehave by deliberately not transmitting the authentication message to a user or failing to provide the authentication message. The "majority" rule is followed by P and CP at the Block-commit stage: the block may be approved by P and CP as long as they get sufficient authentication messages from more than 50% of CS.

3) Block-on-Chain: Block-commit eliminates the necessity for pairwise communication between the CS. Only the P and CP, however, could verify if the $block(i)$ has obtained network-wide approval. CS are unable to reach a consensual result. As a result, the new $block(i)$ cannot be legitimately packed into the blockchain during the first round of consensus. During the second round of consensus, the P transmits the $block(i+1)$ along with the block header that contains the $block(i)$'s hash value. A CP confirms that the hash value of its local $block(i)$ matches the $H(block(i))$ in the block header of $block(i+1)$, as indicated in the Block-request. When a CS gets enough CP-validate messages from P and CP, it can formally broadcast the $block(i)$ that has been approved by the "majority" of CS to the blockchain. It is clear that while the new block in the ETB only needs two rounds of confirmations, the new block in Bitcoin needs six rounds of confirmations to be finally published to the blockchain. The specifics of the original BAC are shown in Algorithm 1 [41]. ($\rightarrow$: broadcasting authenticated messages; $\leftarrow$: receiving authenticated messages) The P calculates the hash value of the $block(i-1)$, represented by $H(I)$, and sends the $block(i)$ to CP. Then CP calculate the hash value of the

Algorithm 1: Block Consistency.

```
Input: $T$: a set of transactions; $t$: $t \in T$; $H()$: the hash
       function; $P = \{P, CP\}$; $C = \{CS\}$
Output: $out$
$T \rightarrow P, C$ ;
while $P$ calculates $H(I) = H(block(i-1))_P$ do
    $block(i) \rightarrow P$;
    $P$ calculate $H(II) = H(block(i-1))_P$;
    if $H(II) \neq H(I)$ then
        $\langle CP - validate \rangle$: $CP(c)_{reject} \rightarrow$ P
    end
    if $H(II) == H(I)$ then
        $\langle CP - validate \rangle$: $CP(c)_{accept} \rightarrow C$
    end
end
while $C \leftarrow \langle CP\text{-}validate \rangle$ do
    $C$ calculate $H(III) = H(block(i-1))_C$ ;
    if $H(III) == H(I)$ then
        $\langle CS - consensus \rangle$: $CS(c)_{accept} \rightarrow P$
    end
end
// the second consensus
if $|CS(c)_{accept}| > |CS(c)_{reject}|$ then
    $block(i+1) \rightarrow CP$
end
while $\langle CP - validate \rangle \rightarrow CS$ do
    $CS$ obey "majority" rule;
    if $CS \leftarrow block(i+1)$ then
        verify $H(block(i))$ in $block(i+1)$;
        if $|CP(c)_{accept}| > \frac{|P|}{2}$ then
            publish $block(i)$ into blockchain
        end
    end
end
out = success
return $out$
```

$block(i-1)$, represented by $H(II)$. If $H(II)$ is not equal to $H(I)$, then CP reject the new $block(i)$ from the P. If $H(II)$ is equal to $H(I)$, then CP accept the new $block(i)$ and broadcast it to C. The C calculate the hash value of the $block(i-1)$ like P, represented by $H(III)$. If $H(III)$ is equal to $H(I)$, then C broadcast their authenticated messages to P, and the second round of consensus starts. The C get the consensus result of the $block(i)$ in the first round by the "majority" rule. If the number of $accepted$ messages exceeds those of $rejected$ ones, then the $block(i)$ can be published into the blockchain.

Thus the time complexity of BAC consensus could be calculated [41]. Suppose there are c CP nodes and n CS nodes in a shard, where c is a fixed constant value and $c \ll n$. The rounds of communication are c and cn in the **Block-request** and CP-validate stage, respectively. The CS-consensus communication's rounds are $n(c+1)$. So the total rounds of communication are $T = 2[c + cn + (c+1)n] = C_1 n + C_2$, where $C_1 = 4c + 2, C_2 = 2c$. So the time complexity of the BAC is $O(n)$.

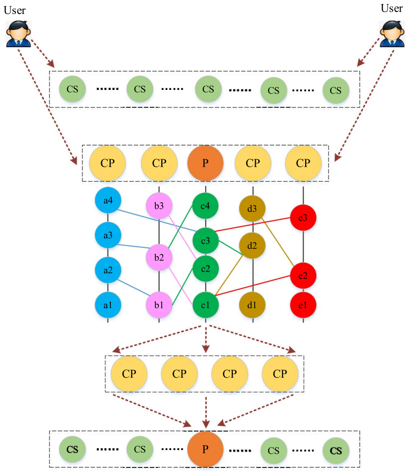


Fig. 5. Improved BAC consensus mechanism for energy trading.

B. *Improved BAC - Asynchronous Byzantine Fault Tolerance*

Although the blockchain is evolving quickly, people are gradually becoming aware of its drawbacks, including those related to fairness, security, and speed, which is the greatest bottleneck. In the blockchain, nodes group transactions into blocks, which are then sequentially linked. The chain storage structure prevents nodes from generating blocks concurrently, slowing down transaction processing. Several situations, such as the dynamic energy market where the price of each transaction varies, do not fit the characteristics of classic blockchains [36]. To officially publish a block on a typical blockchain, it takes at least six blocks (about an hour). In the energy trading industry, where every second thousand transactions might be micro-paid, blockchain is not an appropriate solution [36].

The BAC 2.0 consensus method is depicted in Fig. 5. Directed Acyclic Graph (DAG), an asynchronous data distribution technique, has emerged to enable concurrent writing of transactions, which is a revolution of distributed ledger technology, to increase the efficiency of the blockchain. Baird [42] proposed Hashgraph in 2016. One of the common applications based on the DAG structure is the hashgraph. Through electronic voting, it achieves leaderless Byzantine Fault Tolerance consensus. Hashgraph needs a predetermined amount of nodes in order to adhere to the principle of greater than 2/3 of the total nodes to achieve Byzantine fault tolerance. Hashgraph has a tremendous of use-value because of its special ability to guarantee decentralization without imposing an onerous proof of work requirement. Hashgraph is a revolutionary distributed ledger technology, a promising consensus for energy trading. However, Hashgraph has limitations, such as a fixed number of nodes and being too decentralized for practical applications. In energy trading, different nodes have different roles, and the goal is efficiency, security, and reducing grid load, rather than blindly pursuing decentralization.

The event is an essential element of the ABFT stage. A collection of transactions, timestamps, a hash of references to two parent events, and the use of the event notion rather than the block concept make up the four components of the enhanced

BAC. A hashgraph requires that each event be connected to two parent events, one of which must be the event of a different node and the other of which must be the event of the event itself.

If a child event y is linked to an ancestor event x, then y can see x. If the path goes through the majority of nodes, event y can strongly see the event x. An event is on a new round, designated as R, if it strongly sees the previous events of the majority of nodes. All nodes are consistent in the initial state, which is represented by $R = 1$.

The first event generated in round R is called witness. A witness from round R is considered a famous witness if it is strongly viewed by the majority (more than 2/3) of the witnesses from round $R + 1$. Most consensus algorithms require each node to send votes to each other, resulting in low throughput and poor scalability. In hashgraph, nodes could calculate each other's votes by virtual voting instead of sending votes across the network. "witness" and "round" contain the process of virtual voting. (1) Validity of voting: The procedure of choosing the voting committee members is valid if an event is strongly seen by the majority of witnesses. (2) Aggregate the votes cast for an ancestral event by the voting committee. The ancestral event may be certified as a famous witness, meaning that it cannot be changed, if the number of votes surpasses $2/3$. The consensus cannot be achieved in the $R + 1$ round since the nodes do not know each other's votes. In the $R + 2$ round, the process of counting votes entails gathering the voting outcomes from each node in order to arrive at a consensus.

Our ETB system and BAC algorithm have multiple mechanisms to ensure fairness: whole network verification, incentives (rewards and penalties), sharding technique, and smart contracts. The users in ETB are in the consortium blockchain, and their identity is known and traceable. In addition, there are regulations of government departments and legal means. The detailed analyses of fairness could be found in Appendix B (Supplementary material).

C. Leader Election and Credit Incentive

The election process maintains liveness by allowing the ETB to continue in the event that the P or CP fails. In the event that P or CP fails or makes a mistake, the election procedure shall be applied. CP will communicate the unauthorized block's consensus result to P. As a result, the Block-on-Chain for this unapproved block cannot be completed. If the majority of CP reject the $block(i)$, P's credit score will suffer. As a result, BAC has the advantage of minimizing communication overhead and minimizing the selection of the leading node (the P and CP in ETB). Algorithm 2 illustrates the specifics of the P election and Credit Incentive. When a CP finds the current P is unresponsive, the CP enters the candidate state and starts the election. The CP stops the timer while a block is being executed, but if it is waiting to validate a new block, it restarts the timer. Each CP has a timeout during which it expects to receive a heartbeat from the P. The timeout is reset when a heartbeat is received. If the timer of the P expires, the CP(j) with the highest score becomes the new P and packages the information into blocks. If a CP(j') falsely poses as the new P, the other P will not authenticate it during the "Block-commit" stage of "Block Consistency". The P may also behave maliciously by manipulating data. The CP and C nodes will verify the block body and block height. The P must repackage the block if it receives "majority" rejection messages.

Algorithm 2: P Election.

```
Input: Δ: the timer of CP; s: credit score; HIT:
       Height of the block
Output: CP(j), s
while CP ← block(i) from P do
    start the timer;
    if Δ expires then
        CP(j)'s latest block(i) → P;
        if HIT(block(i')) < HIT(block(i)) then
            CP(j'') ← blocks from CP(n);
            s(P) = s(P)−10;
            s(CP(j)) = s(CP(j))+10;
            s(CP(n)) = s(CP(n))+10
        end
    end
end
return CP(j) and s
```

CP notifies the P of heartbeat messages. If the P's timer expires (it doesn't get a heartbeat from the CP), the procedure of replacing the flawed CP will begin. In accordance with Credit Incentive, if a problematic node CP(j') fails to transmit heartbeat signal, the other CP will deliver the $\langle CP - election$,CP $(j), H(block(i))$, CP(r),CP$(c)\rangle_{\sigma_j}$ message to CP. The P transmits CP(k) from $P(\langle CP - election\rangle)$ to P and CS. The CS with the number k is converted to a CP, and the CP(j') is converted to a CS. If two CS nodes get the same score, the CS with the lowest value is selected to be the new CP node. Along with downtime, the P and CP check the score every x blocks as specified and begin the election process for the CP whose score is less than the threshold. Real-world applications select the suitable value for x. It is estimated to be worth 1,000. To ensure consistency, P and CP are chosen at the expense of ETB availability. Despite the fact that no new blocks will be produced, all nodes will continue to receive client transactions. The CP election and Credit Incentive are shown in Algorithm 3. If the timer of P expires, the P will initiate the replacement of the faulty CP. The P selects CS(k) and sends it to P and CS. The CS with the number k is upgraded to CP, and CP(j') is degraded to CS. The CS with the minimum number is selected to serve as the new CP node if two CS nodes share the highest score.

In terms of the architecture of the method, the number of nodes participating in the consensus process (including the primary node and candidate primary nodes) is predetermined based on the specific requirements of the application. While most consensus nodes (which are typically energy trading users in the real world) can join or leave the network at any time, the total number of consensus nodes is not fixed. Consequently, only the nodes within the committee are required to store the complete blockchain and participate in the consensus process throughout.

Algorithm 3: CP Election.

```
Input: Δ’: the timer of P; $s$: credit score
Output: CS($k$), $s$
if Δ’ expired then
    initiate the election of CP($j'$);
    if $P \leftarrow$ election from $P$ then
        $P \rightarrow \langle$CP-election$\rangle$ to P;
        P selects CS($k$) from $P(\langle$CP-election$\rangle)$;
        CP($j'$) = CS($k$);
        $s$(CP($j'$)) = $s$(CP($j'$))−10;
        $s$(CS($k$)) = $s$(CS($k$))+10;
    end
end
if $x$ blocks committed then
    check on $P$;
    if $s(CP(j'))$ < min $s(P)$ then
        initiate the election of CP($j'$)
    end
end
if $P$’s latest block committed then
    $s$(P($k$)) = $s$(P($k$))+5
end
return $CP(k)$ and $s$
```

In terms of the amount of computation required by the method, both PoW and PoS consume substantial amounts of energy during the consensus process. However, the other consensus mechanisms mentioned in this research consume significantly less computing power, with some requiring almost none.

D. Cross-Shard for Energy Trading Blockchain

The transfer of transactions between different groups of entities is essential to a distributed ledger for energy trading. Cross-shard trading is inevitable and will even be common in energy trading. Entities in energy trading (individual customers, power sector, etc.) may change the shard they are involved, especially in scenarios like electric vehicle energy trading where entities are highly mobile, making secure and effective cross-shard trading particularly crucial.

Some shards may confirm a transaction when processing cross-shard transactions, while others may abort the transaction. We propose an innovative cross-shard processing method applicable to energy trading. Our method is atomic, i.e., a transaction is either committed or aborted. This is done to ensure consistency between energy shards and to prevent individuals or departments within a trading entity from duplicating transactions using inconsistencies across shards. The cross-shard method for P2P energy trading is shown in Fig. 6.

Initialize: Requests for cross-shard transactions are not initiated by the leader node, but by the entity itself. The user initiates authentication requests to the input shards and the output shards.

Blocked: The leader of each input shard verifies the validity of the transaction. If the transaction is valid, the leader node marks the input of the transaction as spent and gossips the $proof\text{-}of\text{-}validity$, which is a Merkle proof of the block containing the transaction. If the transaction is invalid, the leader rejects the cross-shard transaction request, correspondingly generating $proof\text{-}of\text{-}invalidity$. In the actual validation message, a particular bit indicates acceptance or rejection. After all input shards associated with the transaction have verified the request, the user recovers the blocked funds based on proofs sent by a sufficient number of leaders.

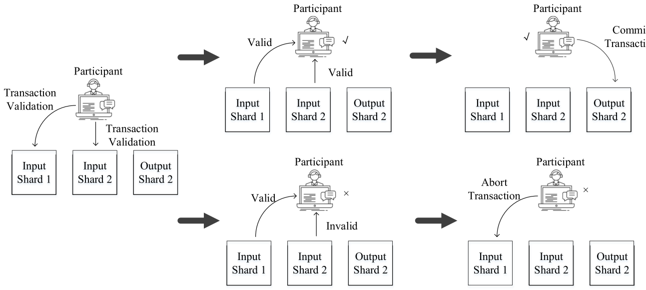


Fig. 6. Cross-shard for P2P energy trading.

Unblocked: Depending on the verification result of the leader in the blocked phase, the user can choose whether to commit or abort the transaction. If all the leader nodes of the input shards associated with the transaction prove that the transaction is valid, the user can gossip the $unblocked\text{-}commit$ message, which contains blocked transaction and all proofs of validity. Correspondingly, the leader node of each output shard associated with the recipient of that transaction validates the transaction and includes it in the next block of the distributed ledger. As long as there is a $proof\text{-}of\text{-}invalidity$ from an input shard, the transaction can not be committed and must be aborted. In this case, the user broadcasts an $unblocked\text{-}abort$ as gossip, asking all leaders of the input shard associated with the transaction to unblock the transaction. Once the $unblocked\text{-}abort$ request is received from the user, the leaders of the input shard again mark the transaction as spendable. Table I shows the list of key abbreviations. Table II shows the list of key symbols. The pricing mechanism based on the Bayesian game of ETB could be found in Appendix A (Supplementary material). The BAC consensus and the cross-shard technique in the ETB help ensure the integrity and scalability of the system while allowing for real-time trading between peers. The proposed ETB for P2P energy trading using Hashgraph is designed for real-time trading between peers, leveraging the fast and secure transaction processing capabilities of Hashgraph’s consensus algorithm and a direct communication channel between buyers and sellers.

V. PERFORMANCE EVALUATION

A Python program and the VIBES blockchain emulator are used to show the proposed BAC consensus technique for ETB. All computations are performed on a Lenovo PC running Windows Ultimate 64-bit, with an Intel i7-8550 CPU running at 1.80 GHz and 8.0 GB of 2133 MHz LPDDR3 memory, as well as Java JDK Version 11.0.10, Scala Version 2.13.5, and Akka Version 2.6.14. Table III compares the BAC 2.0 paradigm to several distributed ledger systems. Our technique achieves immutability, Distributed Denial of Service (DDoS) resistance,

TABLE I
LIST OF KEY ABBREVIATIONS

Abbreviations	**Descriptions**
ABFT	Asynchronous Byzantine Fault Tolerance
BAC	Block Alliance Consensus
BGP	Byzantine Generals Problem
CP	Candidate Primary
CS	Consensus Nodes
CA	Certificate Authority
DAG	Directed Acyclic Graph
DDoS	Distributed Denial of Service
DERs	Distributed Energy Resources
DPoS	Delegated Proof-of-Stake
ETB	Energy Trading Blockchain
IoT	Internet of Things
P	Primary Node
PoW	Proof-of-Work
PoS	Proof-of-Stake
PBFT	Practical Byzantine Fault Tolerance
P2P	Peer-to-Peer

TABLE II
LIST OF KEY SYMBOLS

Symbols	**Descriptions**
P	The set of Primary and Candidate Primary Nodes
C	The set of consensus nodes
$block(i)$	The current unvalidated block at height i
$header(i)$	The block header of the $block(i)$
$data(i)$	The block body of the $block(i)$
$Prev\_Hash$	The hash value of previous $BLOCK(i-1)$
$H()$	The hash function
$CP(j)$	The j-th candidate primary
$N(r)$	The credit score of a node N
$N(c)$	The consensus result of a node N
$\langle m\rangle_{\sigma_N}$	A message m signed by a node N
T	A set of transactions
HIT	The height of a block

fair ordering, fair timestamping, and node dynamicity. The "Dynamicity" in Table III refers to whether the consensus algorithm supports the dynamic addition or deletion of nodes. In a central server architecture, the addition and deletion of nodes are supported, as the server can be configured to accept new connections or terminate existing ones. Some of the leader-based consensus mechanisms support the dynamic addition and deletion of nodes, and some do not. For example, PBFT does not support dynamic changes of nodes, while PoS does. This article uses traditional blockchains for PoW-based blockchain systems, such as Bitcoin. Bitcoin is a public blockchain that allows the dynamic addition and deletion of nodes. Every node in Hashgraph must know about every transaction that has ever occurred on the network, and the order in which those transactions occurred. This requirement makes it difficult to dynamically add or remove nodes from the network. Doing so would require significant coordination and communication among all the existing nodes to ensure that the new node has a complete and consistent view of the network history.

We simulate large-scale blockchain networks with thousands of nodes, making it a useful tool for evaluating the scalability of different blockchain protocols. We also simulate real-world scenarios, such as node failures, network delays, and network partitions, to study the resilience of blockchain networks under adverse conditions. The block size is 2.0 MB, with a propagation latency of 0.8 s. The block size in ABFT stage (micro block size) is 0.5 MB, with a propagation latency of 0.5 s.

Hyperledger Fabric is one of the most widely used blockchain platforms for building decentralized applications and smart contracts. It is a permissioned blockchain, which means that only authorized participants can access the network and perform transactions. Events are initiated by the client, $RecvMsg$ manages the consensus engine upon receipt of events from the client. The $RecvMsg$ function is defined by $Consenter$, which allows the consensus plug-in to receive messages from the network. The Consenter is initialized within $NewConsenter$ ($NewConsenter$ is pluggable). The $Consenter$ provides consensus module methods to invoke the $ExecutionConsumer$ interface, $receiver.ProcessEvent$ is responsible for managing events in the event manager queue, and the $Receiver$ can be plugged. All consensus-related interfaces are all pluggable and provide the corresponding interfaces, custom implementation of $Consenter$ and $Receiver$, $SetReceiver$ at the time of initialization, then can achieve a custom consensus. Algorithm 4 demonstrates the ETB's performance test. We move the detailed descriptions of the node architecture in Hyperledger Fabric to Appendix D (Supplementary material).

A. Blockchain Length and Average Block Time in Energy Trading Blockchain

The experiment was carried out 10 times with the same number of nodes for each approach. The average of these 10 data points from each method is then compared. The duration of the simulation is set to 4 hours. As we test different consensus mechanisms and configurations, we ensure that each experiment produces roughly consistent results with a small standard deviation, indicating the system's stability under test. The "Average Block Time" of PoW, PBFT, improved PBFT (Ref. [22]), BAC, and BAC 2.0 are compared and examined. Hashgraph's "Event" can alternatively be interpreted as "Block". Hashgraph nodes might simultaneously publish blocks. The "Blockchain Length" of PoW, PBFT, improved PBFT (Ref. [22]), DAG (Ref. [38]), Hashgraph, BAC, and BAC 2.0 are also compared and examined. The performance comparison of different consensus mechanisms is shown in Figs. 7 and 8.

"Blockchain Length": The average block duration in a typical blockchain (PoW) is constant, so as the number of nodes increases, the block height essentially stays constant. The PBFT

TABLE III
COMPARISON WITH DIFFERENT LEDGER TECHNOLOGIES FOR P2P ENERGY TRADING NETWORK

Features	**Immutability**	**DoS Resistance**	**Fair Ordering**	**Fair Timestamps**	**Dynamicity**
Central Server	×	×	×	×	✓
Leader Based	✓	×	×	×	\
Traditional Blockchain	✓	✓	×	×	✓
Hashgraph	✓	✓	✓	✓	×
Our model	✓	✓	✓	✓	✓

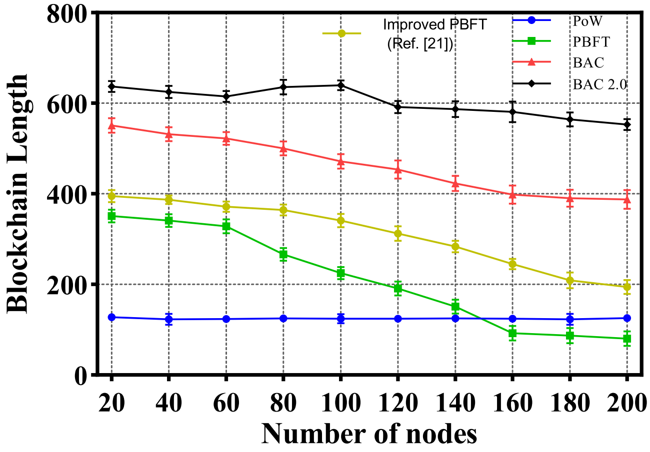


Fig. 7. Blockchain length in ETB platform.

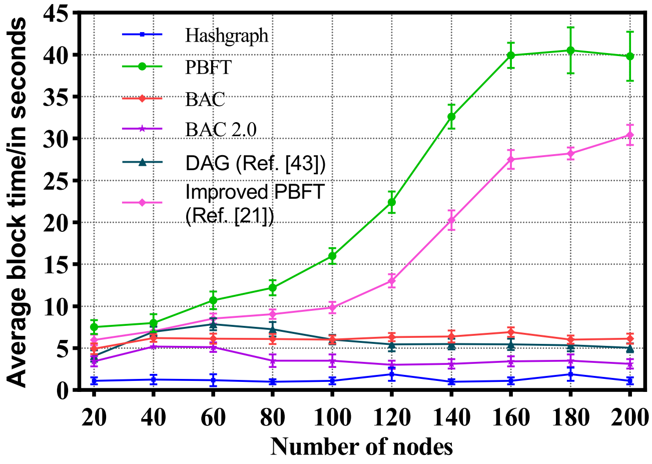


Fig. 8. Average block time in ETB platform.

is not dependent on computing power. In the case of fewer nodes, PBFT has a significantly higher throughput than PoW. However, the communication overhead is much heavier due to the fact that its optimum time complexity is $O(N^2)$. With more than 80 nodes, the blockchain's length drastically shrinks as the number of nodes increases the block's validation latency at each level. Additionally, BAC does not require any processing resources and has a time complexity of $O(N)$. Although BAC requires the confirmation of at least two blocks before producing a new block, the communication overhead is significantly smaller than in PBFT. The longest "Blockchain Length" of the five consensus mechanisms belongs to BAC 2.0. The BAC's throughput is greatly increased using hashgraph (Hashgraph achieves a speed of 50,000 transactions per second in comparison to 15 in Bitcoin, for 30 Ethereum). The block created by the leader P will be distributed directly to the CS in the following blockchain stage after being authenticated by the CP in order to realize distributed storage. Regarding the blockchain length, Hashgraph is not comparable to the other algorithms in this article. Hashgraph uses a parallel block-out approach. Each node can pack the received transactions into a block and publish it. All the other algorithms in this article have transactions packaged by a specific node or committee, and other nodes then verify this block.

"Average Block Time": The average block time doesn't change since the PoW automatically modifies the block difficulty. The behavior and processing efficiency of the node determine the block time. The "Average block time" gradually grows as the number of nodes increases. The growth is relatively smooth when there are only a few nodes. However, the block time increases noticeably when there are more than 80 nodes. Block confirmation in BAC requires two rounds of agreement. However, due of its reduced communication cost and temporal complexity, BAC's "Average Block Time" is still less than PBFT's. The advantage is becoming more apparent, especially when there are more than 100 nodes. The Hashgraph event could be seen as a block in the experiment. The average block time in Hashgraph is the shortest. In the other three approaches, the second block may only be packed after the first block has been validated, but Hashgraph nodes broadcast the block in parallel without verification.

BAC 2.0 itself is stripped from DAG. Compared with DAG, BAC 2.0 can effectively avoid the consistency and conflict problems caused by the non-synchronous consensus mode, thus avoiding double-spending and hard forking. The primary distinction between BAC 2.0 and DAG is that BAC 2.0 uses gossip consistency protocol, an ultimate consistency technique. All nodes will be in the same state eventually, which is a point in time that exists in reality but cannot be proved in theory, even if there is no guarantee that they will all be in the same state at any particular time. The gossip is decentralized, meaning there is no need for a central node, and all nodes are completely

Algorithm 4. ETB Performance Test.

```
Input: numPeers = 1000; numTransactions = 10000
Output: throughput, latency, and scalability

// Initialize ETB netvork
network = initializeNetwork()

// Create multiple participating nodes
for i in range(numPeers) do
    peer = createPeer(i)
    network.addPeer(peer)
end

// Define ETB contract
contract = defineContract()
// Deploy ETB contract
contract.deploy()

// Submit a certain number of transactions
   to the ETB network
for i in range(numTransactions) do
    transaction = createTransaction(i)
    network.submitTransaction(transaction)
end

// Wait for all transactions in the test
   network to be processed
network.waitForTransactions()

// Calculate throughput
throughput = numTransactions /
  network.transactionProcessingTime
// Calculate average latency
latency = network.transactionProcessingTime /
  numTransactions
// Test ETB Scalability
for i in range(numPeers) do
    newPeer = createPeer(numPeers + i)
    network.addPeer(newPeer)
    newTransaction =
      createTransaction(numTransactions + i)
    network.submitTransaction(newTransaction)
      network.waitForTransactions()
end
return Throughput, Latency, and Scalability
```

reciprocal. Hashgraph is more efficient than the gossip employed by Bitcoin. BAC substitutes the Bitcoin system's requirement that nodes reach consensus by broadcasting to the whole network by instead requiring them to transmit messages to a randomly chosen set of nearby nodes. The nodes that receive the messages then repeat the process. This strategy ensures the effectiveness of message processing throughout the network while significantly reducing the bandwidth cost necessary for nodes to establish consensus. The improved PBFT has one less round of intricate mutual communication between nodes than PBFT. It is difficult for participants to agree on a block in a scenario like P2P energy trading because of the unnecessarily high communication complexity of the PBFT.

B. Pending Transactions and Credit Incentive in Energy Trading Platform

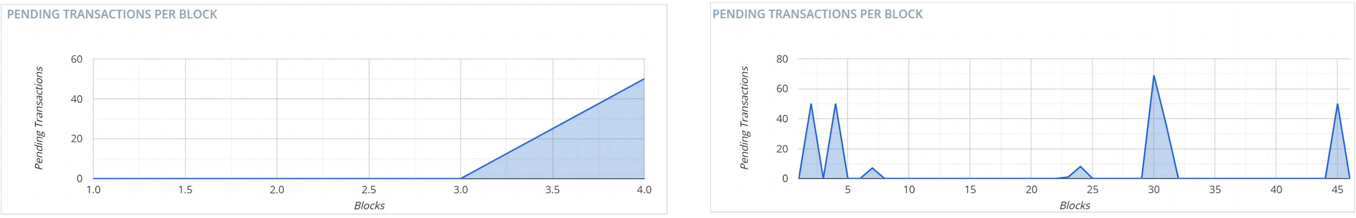
Fig. 9 shows the pending transactions for these three algorithms on the ETB platform. Every block in the blockchain has a transaction pool size that is less than the number of transactions that were previously included in the block when it was created. Because of its complicated interactions and poor scalability, the PBFT creates the most pending transactions. Then we calculate the number of pending PoW transactions as 296, BAC transactions as 43, and BAC 2.0 transactions as 25.

Bitcoin provides two distinct motivations. Mining offers the block reward as well as the transaction charge. To send a transaction, a user must provide the miner a particular amount of Bitcoin. By adding more incentives, the user raises the chance that the miner will include the transaction in the block. In addition, BAC offers two distinct incentives. The blockchain system provides a credit reward based on the efficacy of the preceding block state. The other profit is that CP or P can deliver the appropriate blocks to other nodes. As seen in Fig. 10, a node uses its credits to attach a fee to a transaction in order to encourage the CP to add the transaction in the next block that must be mined. Fig. 10 shows the confirmed transactions with the increase in credit score.

Our proposed approach of fusing Hashgraph and the experimental platform based on Hyperledger Fabric guarantees data privacy. All data transmission in BAC is encrypted; only authorized users can decrypt and access the data. BAC uses digital signature technology to ensure the integrity and authenticity of transactions. Transactions are digitally signed before being created and broadcast to confirm the identity and source of transactions are trusted. In the ABFT stage, BAC is a decentralized technology with no central control point. Therefore, BAC can protect the privacy and security of transaction data, even in the event of a failure or attack, and quickly recover and protect data.

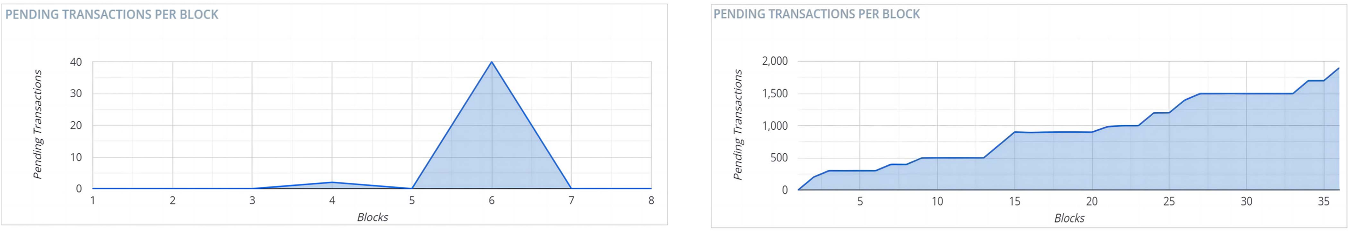
The natural nature of our chosen experimental platform guarantees the security and privacy of the data. The ETB implemented in the Hyperledger Fabric is a permissioned network, meaning only authorized entities can participate and access data. This allows for greater control over the network and data privacy. The ETB uses access control policies to control access to data and network resources. Channels allow for private sub-networks within the larger network, allowing for confidential transactions between specific parties.

In PBFT, each node sends messages to all other nodes in the network, which can result in a high communication burden, especially in large networks. In PoW, nodes compete to solve a mathematical puzzle, and the first node to solve the puzzle broadcasts the solution to the network. This process requires nodes to communicate with each other to verify the validity of the solution. PoW has a high communication burden, especially when the network is congested, leading to increased latency and high energy consumption. PoS requires nodes to hold a certain



Pending Transactions of PBFT

Pending Transactions of PoW



Pending Transactions of BAC

Pending Transactions of BAC 2.0

Fig. 9. Pending transactions.

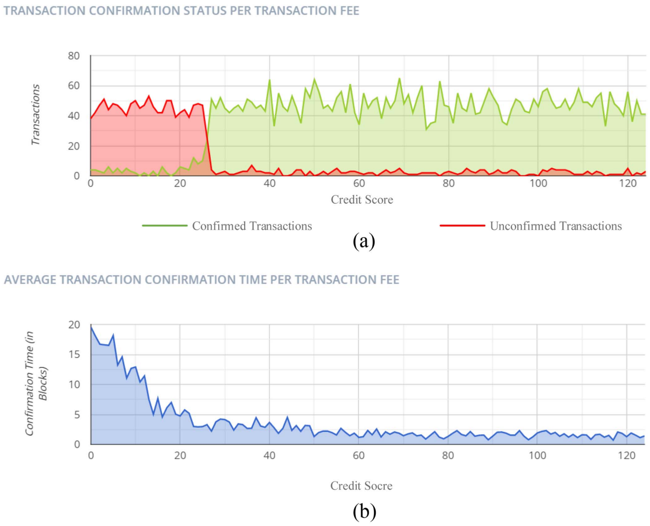


Fig. 10. (a) Confirmed and unconfirmed transactions with the increase of reputation score. (b) Confirmation time with the increase of credit score.

amount of cryptocurrency as collateral to participate in the consensus process. Nodes are selected to validate transactions based on their stake in the network. This process requires less communication than PoW since nodes do not need to compete to solve puzzles.

The BAC utilizes Hashgraph in its ABFT stage. It uses gossip to spread information among nodes. Nodes communicate by randomly selecting other nodes in the network and sharing information with them. This process results in a low communication burden, especially compared to PoW and PBFT. DAG is a consensus mechanism that allows for the parallel processing of transactions, which can result in a lower communication burden than traditional blockchain systems. In DAG, nodes validate transactions by referencing previous transactions, allowing faster confirmation times and higher throughput. However, DAG still requires nodes to communicate with each other to reach a consensus, which can result in a higher communication burden than BAC. Table IV compares different consensus mechanisms for P2P energy trading.

C. The READ and WRITE Performance on the Hyperledger Fabric

Our energy trading blockchain performance under various workloads, ranging from 500 to 2000 TPS, can be seen in Fig. 11. For our ETB system, 1000 users are producing proposals. Fig. 11 illustrates the performance of READ and WRITE and Success Rate. As can be seen in the figure, the throughput of READ can reach the highest to 1056tps at 1200tps workload, 1021tps at 1200tps workload, 980tps at 1400tps, 621tps at 800tps workload, and 659tps at 800tps workload in BAC, Hashgraph, DAG (Ref. Ref. [38]), PBFT, and Improved PBFT (Ref. [22]) respectively, whereas WRITE reaches 936tps at 1200tps workload with a 78% success rate and with 1.14-second latency, 874tps at 1200tps workload with a 72% success rate and with 1.98-second latency, 840tps at 1200tps with a 70% success rate and with 2.67-second latency, 501tps at 600tps workload with a 84% success rate and with 1.69-second latency, and 584tps at 800tps workload with a 70% success rate and with 2.03-second latency in BAC, Hashgraph, DAG, PBFT, and Improved PBFT (Ref. [22]) respectively. Table V shows the Read's and Write's performance test. The first step is to design the network architecture for the Hyperledger Fabric-based ETB system. This involves determining the number of nodes required, their roles and responsibilities, and the communication protocols. The nodes include energy producers, energy consumers, validators, and regulators. Once the network architecture has been designed, the next step is to set up the infrastructure for the ETB system. Smart contracts are implemented on the ETB and executed automatically when certain conditions are met. The user interface is the platform through which energy producers and consumers can participate in the trading process. It should integrate with the Hyperledger Fabric to ensure the seamless execution of energy trading transactions. Once the system is deemed stable and reliable, it can be deployed in a production environment. After the system is

TABLE IV
COMPARISON WITH DIFFERENT CONSENSUS MECHANISMS FOR P2P ENERGY TRADING

Mechanisms	Decentralization	Scalability	Throughput	Latency	Computing	Security
PoW	High	Low	Low	High	High	High
PoS	High	Low	Low	Medium	Medium	Medium
DPoS (Ref. [11])	High	Low	Low	Medium	Medium	Medium
(Improved) PBFT (Ref. [22])	Medium	Low	Low	Medium	Low	Medium
DAG (Ref. [38])	High	Medium	High	Low	Low	Medium
Hashgraph (Ref. [36])	Medium	Medium	High	Low	Low	High
BAC	Medium	High	High	Low	Low	High

TABLE V
PERFORMANCE TEST

Methods	Read		Write	
	Peak (tps)	Success Rate (%)	Peak (tps)	Success Rate (%)
BAC 2.0	**1056**	**88**	**936**	**78**
Hashgraph	1021	72	874	72
DAG	980	70	840	70
PBFT	621	78	501	84
Improved PBFT	659	86	584	70

deployed, it is important to continuously monitor and maintain its performance, security, and reliability.

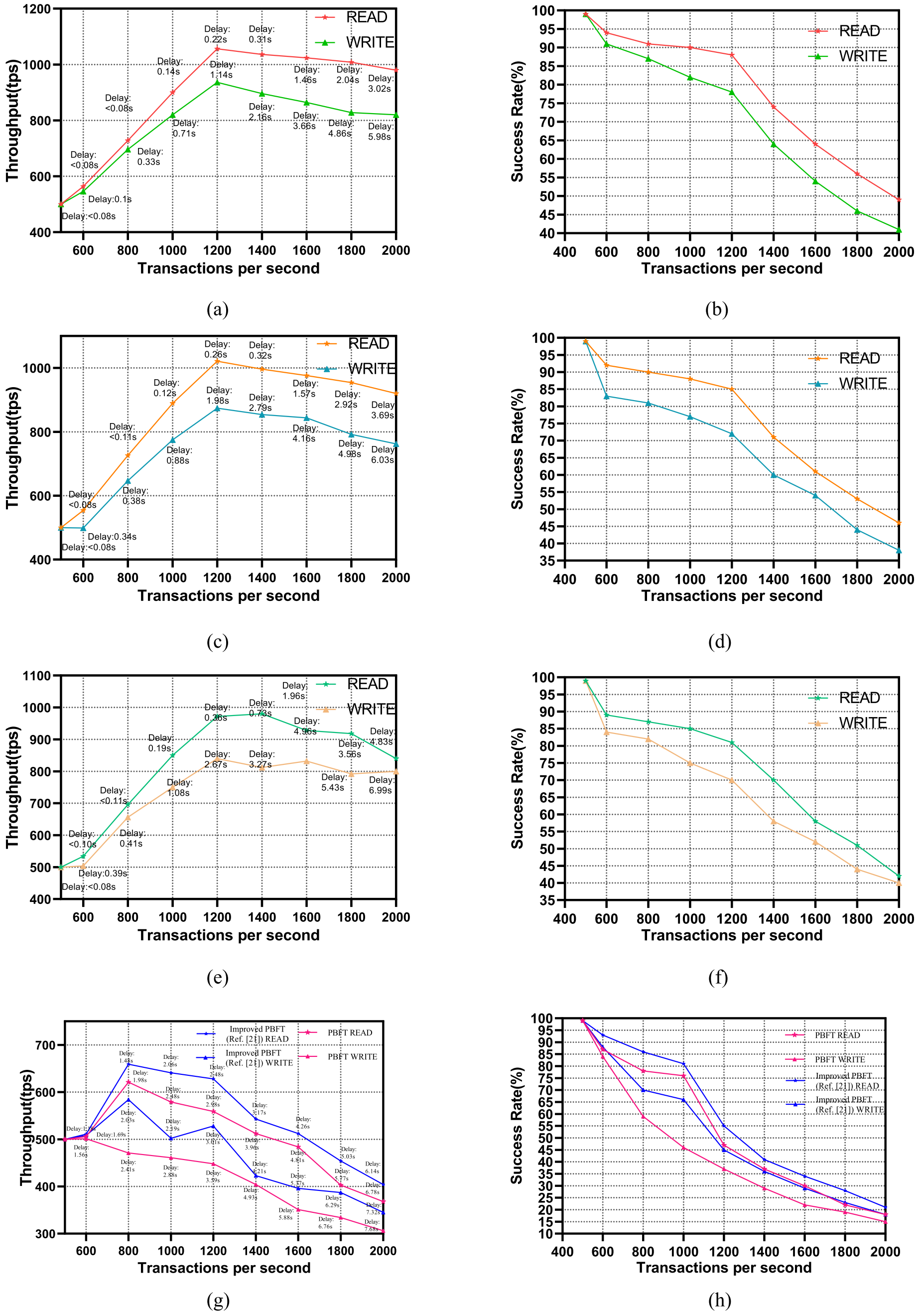
Hashgraph and DAG are two different distributed ledger technologies with distinct performance characteristics. They have different suitability levels for blockchain use cases, including those in consortium or private blockchain networks.

Regarding performance, both Hashgraph and DAG have the potential to achieve high throughput, but they differ in how they achieve it. Hashgraph has a high transaction throughput due to its fast consensus algorithm, which can finalize transactions in near-real-time. In contrast, DAG-based systems can achieve high throughput by allowing multiple transactions to occur in parallel without a central authority, but they may require multiple confirmations to reach a consensus, which slows down the overall transaction processing time.

Regarding latency, Hashgraph typically has lower latency than DAG-based systems, as it can finalize transactions in a matter of seconds. In comparison, DAG-based systems may require multiple rounds of confirmations, which takes longer. In terms of suitability for a consortium or private blockchain network, Hashgraph may be more appropriate due to its strong security guarantees and efficient consensus algorithm, which can facilitate trust between a group of known participants. DAG-based systems, while more scalable, may be less secure and more difficult to govern in a consortium setting due to the lack of a central authority.

Hashgraph utilizes virtual voting and gossip as its key approaches to create a quick, safe, and equitable system. The user i propagates some event I to user j, after which user j broadcasts I to user f, accompanied by the new event that user j itself wants to inform user f about. Hashgraph uses virtual voting and gossip as its main strategies to establish a rapid, secure, and fair system. The propagation of all propagation histories, etc. The ability to conduct "virtual voting" (visible voting on already known events without network propagation) will be made possible once network propagation has been completed since all nodes will have learned which events each node carries. Hashgraph uses a small amount of total bandwidth, making it quick and high throughput. Hashgraph uses gossip to effectively use idle network resources across several nodes, however as a result, network communication volume per node skyrockets while network performance is only marginally improved.

Even if Hashgraph does not have to contain the whole history of the ledger in the message, over time, the length of each message will ultimately reach a linear order of magnitude of the number of participants. Even if there are fewer propagated events, a newly wrapped event might still have considerable content. BAC limits the number of people actually involved in block genesis and verification to a single shard. The main benefit of doing this is that Hashgraph scales for throughput but not for nodes, and BAC will be better suited for P2P energy trading scenarios because Hashgraph has extremely high hardware requirements, and it is a significant overhead for regular participants to execute commands in the ledger, maintain gossip charts, and implement virtual voting mechanisms.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 11. Performance of READ and WRITE and success rate. (a) BAC 2.0's throughput. (b) BAC 2.0's success rate. (c) Hashgraph's throughput. (d) Hashgraph's success rate. (e) DAG (Ref. [38])'s throughput. (f) DAG (Ref. [38])'s success rate. (g) PBFT's and improved PBFT (Ref. [21])'s throughput. (h) PBFT's and improved PBFT (Ref. [21])'s success rate.

VI. CONCLUSION AND THE ROAD AHEAD

In this work, we propose an ETB for P2P energy trading based on blockchain technology. By eliminating centralized third-party systems, the ETB performs more efficiently. We propose a novel BAC consensus breaking through the impossible triangle of the ETB. Our improved BAC has an average blockchain length that is 4.84 times more than PoW, 2.85 times greater than PBFT, 1.94 times greater than improved PBFT, and 1.30 times greater than the original BAC. The improved BAC has an average latency that is 84% less than PBFT, 77% less than improved PBFT, 37% less than DAG, and 39% less than the original BAC. Our ETB's READ performance can achieve the most outstanding throughput of 1056 tps at a workload of 1200 tps, while WRITE can achieve 936 tps at a workload of 1200 tps with a success rate of 78% and 1.14 seconds of latency. Implementing the ETB system requires a combination of technical, business, and regulatory expertise. It involves a complex set of activities that need to be carefully planned and executed. Much work needs to be conducted to develop the ETB system and the BAC consensus mechanism. For future work, we will elect the committee and primary nodes from a cryptographic perspective. Although our work solves the two defects of Hashgraph, the problem of transaction duplication in Hashgraph is still not solved, which may be solved from the perspective of timestamp and random number.

ACKNOWLEDGMENT

The preliminary work for this article was presented at the 2022 IEEE Industry Applications Society Annual Meeting [1].

Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions that have helped improve this article substantially.

REFERENCES

[1] Y. Wang et al., “A fast and secured peer-to-peer energy trading using blockchain consensus,” in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 2022, pp. 1–8.

[2] P. Siano, G. De Marco, A. Rolán, and V. Loia, “A survey and evaluation of the potentials of distributed ledger technology for Peer-to-Peer transactive energy exchanges in local energy markets,” *IEEE Syst. J.*, vol. 13, no. 3, pp. 3454–3466, Sep. 2019.

[3] M. Lotfi, C. Monteiro, M. S. Javadi, M. Shafie-Khah, and J. P. Catalão, “Optimal prosumer scheduling in transactive energy networks based on energy value signals,” in *Proc. IEEE Int. Conf. Smart Energy Syst. Technol.*, 2019, pp. 1–6.

[4] F. R. Badal, P. Das, S. K. Sarker, and S. K. Das, “A survey on control issues in renewable energy integration and microgrid,” *Protection Control Modern Power Syst.*, vol. 4, no. 1, pp. 1–27, 2019.

[5] M. H. Rehmani, M. Reisslein, A. Rachedi, M. Erol-Kantarci, and M. Radenkovic, “Integrating renewable energy resources into the smart grid: Recent developments in information and communication technologies,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 7, pp. 2814–2825, Jul. 2018.

[6] M. S. Javadi, A. E. Nezhad, A. R. Jordehi, M. Gough, S. F. Santos, and J. P. Catalão, “Transactive energy framework in multi-carrier energy hubs: A fully decentralized model,” *Energy*, vol. 238, 2022, Art. no. 121717.

[7] L. Che, M. Shahidehpour, A. Alabdulwahab, and Y. Al-Turki, “Hierarchical coordination of a community microgrid with AC and DC microgrids,” *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3042–3051, Nov. 2015.

[8] M. Z. Gunduz and R. Das, “Cyber-security on smart grid: Threats and potential solutions,” *Comput. Netw.*, vol. 169, 2020, Art. no. 107094.

[9] S. N. Islam, Z. Baig, and S. Zeadally, “Physical layer security for the smart grid: Vulnerabilities, threats, and countermeasures,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6522–6530, Dec. 2019.

[10] M. B. Mollah et al., “Blockchain for future smart grid: A comprehensive survey,” *IEEE Internet Things J.*, vol. 8, no. 1, pp. 18–43, Jan. 2021.

[11] J. Chen et al., “Collaborative trust blockchain based unbiased control transfer mechanism for industrial automation,” *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4478–4488, Jul./Aug. 2020.

[12] W. Gao, W. G. Hatcher, and W. Yu, “A survey of blockchain: Techniques, applications, and challenges,” in *Proc. IEEE 27th Int. Conf. Comput. Commun. Netw.*, 2018, pp. 1–11.

[13] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, and M. Imran, “Blockchain for cloud exchange: A survey,” *Comput. Elect. Eng.*, vol. 81, 2020, Art. no. 106526.

[14] N. Kshetri, “Can blockchain strengthen the Internet of Things?,” *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[15] Z. Che, Y. Wang, J. Zhao, Y. Qiang, Y. Ma, and J. Liu, “A distributed energy trading authentication mechanism based on a consortium blockchain,” *Energies*, vol. 12, no. 15, 2019, Art. no. 2878.

[16] M. Salimitari, M. Chatterjee, M. Yuksel, and E. Pasiliao, “Profit maximization for Bitcoin pool mining: A prospect theoretic approach,” in *Proc. IEEE 3rd Int. Conf. Collaboration Internet Comput.*, 2017, pp. 267–274.

[17] W. Zhao et al., “Consortium blockchain-based microgrid market transaction research,” *Energies*, vol. 12, no. 20, 2019, Art. no. 3812.

[18] G. Wood et al., “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32.

[19] J. Xie et al., “A survey of blockchain technology applied to smart cities: Research issues and challenges,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 3, pp. 2794–2830, thirdquarter 2019.

[20] H. Huang, W. Kong, S. Zhou, Z. Zheng, and S. Guo, “A survey of state-of-the-art on blockchains: Theories, modelings, and tools,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–42, 2021.

[21] C. Zhang, L. Zhu, C. Xu, J. Ni, C. Huang, and X. S. Shen, “Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4410–4425, Dec. 2022, doi: 10.1109/TMC.2021.3080714.

[22] H. N. Abishu, A. M. Seid, Y. H. Yacob, T. Ayall, G. Sun, and G. Liu, “Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the Internet of Electric Vehicles,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 946–960, Jan. 2022.

[23] M. Salimitari, M. Chatterjee, and Y. P. Fallah, “A survey on consensus methods in blockchain for resource-constrained IoT networks,” *Internet Things*, vol. 11, 2020, Art. no. 100212.

[24] M. L. Di Silvestre et al., “Ancillary services in the energy blockchain for microgrids,” *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 7310–7319, Nov./Dec. 2019.

[25] M. Gough et al., “Blockchain-based transactive energy framework for connected virtual power plants,” *IEEE Trans. Ind. Appl.*, vol. 58, no. 1, pp. 986–995, Jan./Feb. 2022.

[26] M. Gough et al., “Development of a blockchain-based energy trading scheme for prosumers,” in *Proc. IEEE Madrid PowerTech*, 2021, pp. 1–6.

[27] E. E. Santos-Gonzalez, G. Gutierrez-Alcaraz, A. E. Nezhad, M. S. Javadi, G. J. Osório, and J. P. S. Catalão, “Stochastic distribution network operation for transactive energy markets,” in *Proc. IEEE Int. Conf. Environ. Elect. Eng., IEEE Ind. Commercial Power Syst. Europe*, 2021, pp. 1–6.

[28] L. Lamport, R. Shostak, and M. Pease, “The byzantine generals problem,” in *Proc. Concurrency: Works Leslie Lamport*, 2019, pp. 203–226.

[29] M. Castro et al., “Practical byzantine fault tolerance,” in *Proc. 3rd Symp. Oper. Syst. Des. Implementation*, 1999, pp. 173–186.

[30] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, “Zyzzyva: Speculative byzantine fault tolerance,” in *Proc. 21st ACM SIGOPS Symp. Oper. Syst. Princ.*, 2007, pp. 45–58.

[31] A. Sheikh, V. Kamuni, A. Urooj, S. Wagh, N. Singh, and D. Patel, “Secured energy trading using byzantine-based blockchain consensus,” *IEEE Access*, vol. 8, pp. 8554–8571, 2020.

[32] L. Feng, H. Zhang, Y. Chen, and L. Lou, “Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain,” *Appl. Sci.*, vol. 8, no. 10, 2018, Art. no. 1919.

[33] Y. Wang et al., “Study of blockchains’s consensus mechanism based on credit,” *IEEE Access*, vol. 7, pp. 10224–10231, 2019.

[34] D. Wang and L. Wang, “Consensus mechanism of multi-energy interactive subject based on practical byzantine fault tolerance algorithm,” *Automat. Elect. Power Syst.*, vol. 43, no. 09, pp. 41–49, 2019.

[35] W. Cai, W. Jiang, K. Xie, Y. Zhu, Y. Liu, and T. Shen, “Dynamic reputation–based consensus mechanism: Real-time transactions for energy blockchain,” *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 3, 2020, Art. no. 1550147720907335.

[36] G. Bansal and A. Bhatia, “A fast, secure and distributed consensus mechanism for energy trading among vehicles using hashgraph,” in *Proc. IEEE Int. Conf. Inf. Netw.*, 2020, pp. 772–777.

[37] J. García-Hernández, L. G. Marín-Collazos, G. Jiménez-Estévez, and P. Mendoza-Araya, “Distributed ledger technologies based microgrid energy management using IOTA tangle,” in *Proc. IEEE CHILEAN Conf. Elect., Electron. Eng., Inf. Commun. Technol.*, 2021, pp. 1–5.

[38] B. Wang, M. Dabbaghjamanesh, A. Kavousi-Fard, and S. Mehraeen, “Cybersecurity enhancement of power trading within the networked microgrids based on blockchain and directed acyclic graph approach,” *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 7300–7309, Nov./Dec. 2019.

[39] S. Nakamoto, “Bitcoin: A Peer-to-Peer electronic cash system,” *Decentralized Bus. Rev.*, vol. 4, no. 2, 2008, Art. no. 21260.

[40] C. Liu and Z. Li, “Comparison of centralized and Peer-to-Peer decentralized market designs for community markets,” *IEEE Trans. Ind. Appl.*, vol. 58, no. 1, pp. 67–77, Jan./Feb. 2022.

[41] Y. Wang, Y. Ma, Y. Qiang, J. Zhao, Y. Li, and K. Li, “BAC: A block alliance consensus mechanism for the mine consortium blockchain,” *Concurrency Comput.: Pract. Experience*, vol. 34, no. 27, 2022, Art. no. e7344.

[42] L. Baird, “The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance,” Swirlds, Dallas, TX, USA, Tech. Rep. SWIRLDS-TR-2016-01, 2016.