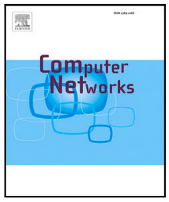
Computer Networks 251 (2024) 110635

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Lightweight adaptive Byzantine fault tolerant consensus algorithm for distributed energy trading

Jin Ye [a,b,*], Huilin Hu [a], Jiahua Liang [a,b], Linfei Yin [c], Jiawen Kang [d]

[a] School of Computer and Electronic Information, Guangxi University, Nanning, 530004, China
[b] Guangxi Key Laboratory of Multimedia Transmission and Network Technology, Nanning, China
[c] School of Electrical Engineering, Guangxi University, Nanning, 530004, China
[d] School of Automation, Guangdong University of Technology, Guangzhou, 510145, China

ARTICLE INFO

Keywords:
Blockchain
Consensus algorithm
Energy trading
Reputation calculation
Byzantine fault tolerance

ABSTRACT

With the rapid development of smart grid, constructing distributed energy trading market (DETM) based on blockchain to coordinate distributed energy resources (DER) has become a future direction. However, existing consensus algorithms of blockchain face many challenges in large-scale energy trading scenarios, such as high resource overhead, slow transaction procedure. To solve the above crucial problems for wide deployment of distributed energy trading, this study proposes a novel consensus algorithm named Lightweight adaptive Byzantine fault tolerant consensus (LA-BFT), and a reputation calculation method based on behavioral characteristics for selection of consensus nodes. The LA-BFT consists of two parts: (i) weak consensus for normal cases. By introducing threshold signature mechanism, weak consensus simplifies the consensus process to achieve linear communication complexity $O(n)$. (ii) byzantine node detection scheme is enable for malicious cases. With consensus committee, the detection scheme can detect the potential byzantine nodes by cross-validation, which ensures transaction safety. The reputation calculation method is presented to cooperate with LA-BFT to elect leaders and candidates for consensus procedure. Once a round of consensus is completed, the reputation of each node needs to be updated, only nodes with high reputation are eligible to become leaders or committee nodes in the next round. With the reputation calculation method, honest nodes and byzantine nodes can be effectively identified, ensuring the security of the consensus process. Numerical results indicate that LA-BFT exhibits superior performance on communication overhead and bandwidth occupancy in large-scale concurrent energy trading scenarios. When the number of nodes is 50, under normal scenarios, LA-BFT's communication overhead is notably lower, constituting a mere 6.02% of PBFT and 24.26% of SHBFT, while bandwidth occupancy amounts to merely 5.55% of PBFT and 9.76% of SHBFT.

1. Introduction

Following the rapid development of power system technology, an increasing number of distributed energy resources (DERs) are being integrated into power grid [1]. The power supply paradigm is shifting from centralized to decentralized [2]. Since DERs is mainly composed of photovoltaic arrays and wind turbine set, featuring randomness, intermittency, and volatility, integrating them into distributed networks is likely to have an unpredictable effect on the grid [3]. In this background, microgrids have emerged as a means of coordinating DERs for the purpose of point-to-point energy transmission and local consumption within a specific area [4]. This approach not only isolates the shocks of power grid originate from DERs [5], but also improves the flexibility and scalability of distributed system [6].

To further improve the energy utilization as well as maximize the benefit of participants in microgrid, a practical and effective solution is to establish distributed energy trading market (DETM) [7,8]. Participants among DERs are allowed to negotiate with each other on quantity and price for energy transaction in DETM, following which the distributed system operator shall dispatch energy transmission in accordance with the negotiation outcomes [9]. For example, TeMix is a energy trading platform in the United States, providing energy trading and distributed management services. In Germany, Peer Energy Cloud is a virtual market based on commercial cloud, which has facilitated local energy transaction. In the UK, Piclo is an independent trading market, providing a platform for renewable energy, batteries and electric vehicles to trade energy as and when needed.

* Corresponding author at: School of Computer and Electronic Information, Guangxi University, Nanning, 530004, China.
E-mail addresses: yejin@gxu.edu.cn (J. Ye), 2313394010@st.gxu.edu.cn (H. Hu), liangjh23@126.com (J. Liang), yinlinfei@gxu.edu.cn (L. Yin), kavinkang@gdut.edu.cn (J. Kang).

https://doi.org/10.1016/j.comnet.2024.110635
Received 24 January 2024; Received in revised form 21 May 2024; Accepted 4 July 2024
Available online 8 July 2024

Despite some successes in the development of DETM, there still are many challenges in the large-scale promotion of DETM. To begin with, the energy transaction between DERs is featured by small amount and high frequency. As the number of DERs increases, the operational pressure on traditional centralized transaction centers increases as well [10]. Besides, a crisis of confidence among market participants has been observed, as some dishonest participants have taken advantage of fraudulent means to obtain improper gains, thereby adversely impacting the fairness and security of transactions [11]. What is more, the traditional centralized management mode is vulnerable to hacker attacks, which can easily lead to single-point failure [12] or even information leakage [13], posing a potential risk to user privacy and security.

In recent years, the emergence of blockchain technology has proposed a new solution to above DETM problems [14]. Blockchain is essentially a new distributed database with chain structure, which brings about a shared ledger [15]. Because of decentralized, immutable and traceable, blockchain has been implemented in many fields now, such as energy trading and so on. In literature [16], in order to solve the single point of failure problem, the authors propose a distributed DM architecture based on blockchain and expound a replicated redundancy Practical Byzantine Fault Tolerance algorithm to increase the security of the system. In literature [17], blockchain is applied to solve the single point of failure problem in the P2P energy trading market and ensure safety of two side auction in trusted transaction environment. In literature [18], a novel blockchain-based secure access framework (BSAF) for cloud-device service collaborations with privacy protection is proposed and two smart contracts are designed: a request verification smart contract to verify the access rights of users, and a behavior punishment smart contract to audit users' access behaviors. In literature [19], a two-layer distributed P2P energy trading platform is proposed. The market layer is responsible for the settlement of auction pool, while the blockchain layer employs smart contracts to provide secure and reliable transactions service for the platform. The above research is a theoretical study of the application of blockchain in various fields including energy trading.

There is no doubt that blockchain brings a new solution to the problem of energy trading. In the energy trading scenario, consensus algorithms commonly used at present include two types, namely consensus with mining and consensus with communication [20]. However, distributed energy trading is usually auctioned in cycles, and if the transaction initiated within the cycle cannot be confirmed by consensus, the user who initiated the transaction will lose the economic benefits that are expected to be obtained during the cycle exchange, so that the enthusiasm of users to participate in the transaction is hit. Due to different factors such as market size, the time span t of a single cycle is also different, for example, in literature [21], the trading period t is defined as 10 minutes. Regardless of the value of t, processing concurrent transactions of large-scale DER in a limited time t is still a huge challenge for existing consensus algorithms. Specifically, PoW and PoS algorithms, as consensus algorithms with mining, have high resource consumption, low algorithm efficiency, it is difficult to ensure that transactions are completed in a limited time, and poor adaptability in distributed energy trading scenarios [20]. In addition, among the commonly used consensus algorithms with communication, Raft algorithm is highly efficient, but it is only applicable to ideal scenarios that are completely trusted and cannot tolerate Byzantine attacks, nor can it be applied to actual energy trading scenarios [22]. Although PBFT consensus can tolerate Byzantine nodes [23], with the increasing of node size, its performance will decline sharply, the consensus efficiency will be affected, and lead to huge delays, and it is difficult to ensure that large-scale transactions can be processed in a limited time. To sum up, blockchain-based consensus algorithms also have some challenges that cannot be ignored, the most urgent of which is the resource overhead that leads to low throughput and limited scale [24]. Therefore, this work focus on exploring a consensus algorithm to support large-scale distributed energy trading.

The main contributions of this work:

(1) The communication complexity of traditional byzantine fault tolerant consensus algorithms is $O(n^2)$, which are not suitable for large-scale systems. Nevertheless, this work proposes a Lightweight adaptive Byzantine fault tolerant consensus (LA-BFT) algorithm for distributed energy trading, whose time complexity is merely $O(n)$ under normal scenario. And the committee is established to cross-check the transaction requests received by the community in four stages to locate Byzantine nodes, with time complexity of $O(Dn)$, where D is the number of committee members. Totally, our algorithm consistently demonstrates lower communication overhead.
(2) A reputation calculation is proposed to work with LA-BFT. The reputation value of communities is updated according to multi-dimensional mutual evaluation, which is utilized for the selection of the next round's leaders and committee nodes, ensuring the security of the consensus process.
(3) The performance of LA-BFT is evaluated by simulation experiments. Results show that the proposed LA-BFT has significant advantages in terms of communication overhead, bandwidth usage, consensus delay, throughput, and transaction failure rate in large-scale concurrent scenarios.

The structure of the remaining part of this work is as follows: the next section is related work of consensus algorithm in Energy Blockchain and some existing problems. Section 3 is our lightweight adaptive byzantine fault tolerant consensus algorithm and security analysis of LA-BFT. Section 4 is the performance of LA-BFT and Section 5 is conclusion.

2. Related work

Consensus algorithm is a crucial component in blockchain, which enables nodes cooperate together without centralized management and third-party supervision to make blockchain a self-organizing decentralized network. However, the insufficient efficiency of consensus algorithm has become the main reason limiting the system expansion. At present, consensus algorithms applied in distributed energy trading can be categorized by the following two categories: consensus with mining and consensus with communication.

2.1. Consensus with mining

As for consensus with mining, network nodes need to package transaction information into blocks according to certain rules. This process requires large computing resources of network nodes, which is called mining. At the same time, the nodes which do mining are called miners. Those miners consume vast amounts of computing resources (including electricity) to compete for the rights to append a new block to current blockchain and obtain some reward such as digital coins.

In literature [25], a proof of work (PoW) consensus algorithm cost is considered in energy transaction platforms. The study proves that the total cost of participants is lower than traditional market structure. In literature [23], an optimized consensus named Proof of Stake (PoS) is introduced to alleviate mining cost and electricity consumption of PoW in distributed P2P energy trading market. Compared to PoW, PoS has improved the algorithm efficiency, but the resource consumption still cannot be ignored, leading to a possible large delay of consensus when there are a large number of participants, making it difficult to ensure that transactions are completed in a limited time [26]. In addition, miners with high holding equity are more likely to obtain rights of appending new blocks, resulting in miners with low equity are less willing to participate in consensus, leading to the Matthew effect [27]. In literature [28], a PoS-like consensus based on Proof of Credit Stake (PoCS) is proposed. By introducing credit mechanism, the higher credit value of the node is, the greater probability of obtaining the rights of appending new blocks. In this way, DETM can encourage participants act honestly in the market transaction.

Table 1
Comparison of different consensus algorithms for distributed energy trading.

	PoW	PoS	Raft	PBFT	LA-BFT
Mining	Yes	Yes	No	No	No
Communication complexity	–	–	$O(n)$	$O(n^2)$	$O(n)$
Security	51% attack	51% attack	Nonsupport	$n \geq 3f+1$	$n \geq 3f+1$
Drawback	Waste of resources	Matthew Effect	Poor adaptability	High latency	–

Note: f stands for the number of malicious nodes and n for the number of summary points

In literature [29], a proof of cooperation (PoC) consensus algorithm is proposed. By defining co-operation factor, PoC elect miners based on the factor and the quantity of BEUs in their wallet accounts. This method is more suitable for multi-agent systems of energy trading applications than traditional PoW algorithm. In literatures [30], a proof of authority (PoA) consensus algorithm is utilized to support the energy trading. The PoA consensus algorithm designates a limited number of authority nodes as miners, which greatly saves computing resources. However, the decentralization degree of PoA is low, which is not suitable for the real distributed energy trading environment.

2.2. Consensus with communication

As for consensus with communication, this kind of algorithm achieves consensus through messages communication rather than consume meaningless computing resources like consensus with mining. In literature [31], a secure data aggregation based on homomorphic encryption and the practical byzantine fault tolerance (PBFT) consensus is proposed for microgrids. PBFT requires a large number of broadcast and verification messages in the process, resulting in a high cost of communication. But the transaction delay depends on communication delay between nodes, which is much smaller than that of consensus with mining. In literature [32], a PBFT-like consensus algorithm is adopted to a hierarchical energy trading framework, in which byzantine nodes are detected through the mutual verification of committee nodes. Once byzantine nodes being confirmed and blacklisted, they are forbidden from participating in market transactions. However, in this consensus process, a large number of message broadcast will be induced. What is more, the market clearing price of DETM is also need several iterations with a large amount of time cost for calculation.

In literature [33], the existing consensus algorithms of energy trading are compared, among which the Raft consensus algorithm performs well in terms of latency and throughput. However, it cannot tolerate byzantine fault that means it is only suitable for ideal scenario instead of real distributed energy trading market. In literature [34], a privacy-preserving-BFT-based (PP-BFT-based) coordination algorithm is proposed to ensure the correctness of trading results for energy trading. Compared with PBFT, PPBFT has more privacy protection features, but it also reduces the efficiency by about 30%. In literature [29], a Peer-to-Peer energy trading architecture based on istanbul byzantine fault tolerance (IBFT) consensus algorithm is proposed. The IBFT simplifies the consensus process on the basis of PBFT, but still has high communication complexity. In literature [35], a delegated practical byzantine fault tolerance (DPBFT) consensus algorithm is proposed to improve the efficiency of energy trading markets system.

2.3. Brief summary

Table 1 lists the performance of different consensus algorithms. It can be seen that although existing consensus with mining (such as PoW and PoS) have excellent security which can against 51%-attack, they have to consume a large amount of computing resources, resulting in high consensus delay.

The consensus with communication can be further divided into byzantine fault intolerable and byzantine fault tolerable. A typical case of byzantine fault intolerable is Raft algorithm, which has superior performance in consensus latency and throughput. However, Raft is difficult to be applied to real energy trading scenarios because Raft cannot tolerate byzantine fault. Therefore, Raft is rarely mentioned in the current researches. A typical case of byzantine fault tolerable is PBFT algorithm, which requires network nodes to achieve consensus through a large amount of communication, is proved to be secure when the proportion of malicious nodes is less than $1/3$ and efficiency is higher than PoW and PoS. But exponential communication complexity of PBFT also costs a mount of network bandwidth which may cause potential network congestion.

What should be emphasized is that the efficiency and security of consensus have important impact on transaction completion ratio. In distributed energy trading, participants are constrained to only submit bids within designated trading rounds, each of which lasts for a certain duration, such as 10 minutes in some scenarios. When there are a large amount of participants simultaneously request for trade, if the consensus algorithm cannot deal with concurrent requests in a short time, most of the trades will be failed, resulting in the revenue of participants decrease sharply comparing what they should deserve in the transaction round. What is worse, this problem is more prominent in the large-scale energy trading scenario. This work aims to simplify the consensus procedure so that transaction delay and throughput performance of consensus algorithm can be improved in large-scale DETM.

3. Distributed energy trading framework

This work aims to improve the performance of distributed energy trading system by resource-efficient consensus algorithm which is named as LA-BFT. This section describes the detailed design of distributed energy trading framework based on LA-BFT.

3.1. System model

Microgrid is consists of multiple DERs whose role includes producers and consumers, in which the generative energy can be transported from producers to consumers, lead to supply and demand balance among DERs. However, there are still situations that supply and demand imbalance such as energy surplus overflow or energy shortage in the whole microgrid. To further improve the energy utilization of DERs, multi-microgrid with energy trading agents are considered to be connected as a Energy Blockchain network, which complete the energy transaction across different microgrids. The main components of Energy Blockchain network are shown in Fig. 1.

Community: It is a microgrid in a small area. Community integrates microturbine (MT), wind turbine (WT), photovoltaic (PV) system, energy storage system (ESS), responsive load (RL), electric vehicle (EV) and other DERs. A community can be either a producer or consumer based on different situations. For instance, when there is energy surplus, it can sell energy to other communities. On the contrary, when energy load exceeds the supply capacity, it can purchase energy from other communities.

In addition, the communities participates in consensus process as blockchain nodes. The blockchain nodes are divided into general communities and committee members. The committee members are composed of two categories: leader and candidates. Leader is only one elected by the committee members, that is responsible for collecting voting messages and aggregating communities' signature shares. Details

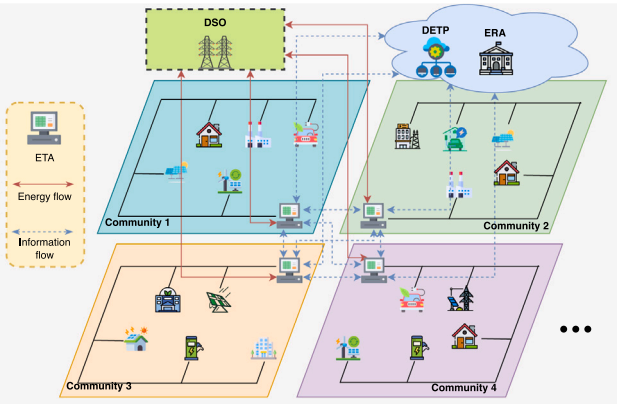


Fig. 1. Distributed energy trading framework.

are described in Section 3.2.1. Candidate is specially elected as a superior to participate the byzantine nodes detection scheme. Once the nodes being judged as byzantine node, they are prevented from participating consensus process. Details are described in Section 3.2.2.

Distributed energy trading provider (DETP) : It provides blockchain-based DETM services to various communities, such as decentralized application of trading platform and access control of Energy Blockchain network. It is also blockchain system maintainer. When a community requests is permitted to access to DETP, it becomes a node in Energy Blockchain network and be allowed to participate in transaction and consensus in DETM.

Energy trading agent (ETA) : It is virtual agent for community to participate in DETM. When the community has demand to purchase/sell energy, the transaction request will be initiated by ETA to Energy Blockchain network.

Distributed system operator (DSO) : It is a authority that is responsible to schedule the energy among DERs. As a light node of Energy Blockchain network, DSO only receives new blocks generated from a completed consensus rather than participate in process of consensus. According to the transaction information in the blocks, DSO dispatches the matched DERs to transfer energy.

Energy regulatory authority (ERA) : It is a credible energy regulator. The energy trade has to be regulated by policies, such as tracing the energy flows or monitoring carbon emissions. In this framework, the ERA is also responsible to elect the leader from candidates and maintain the reputation of each nodes based on their behavior in consensus. Details are described in Section 3.3

In normal cases, the system executes weak consensus process, in which signature shares are aggregated which could be verified by public key. If verification fails in preceding steps that indicates that the byzantine node act maliciously in consensus then the byzantine node detection scheme will be triggered. Hence, the committee members will detect byzantine nodes through the cross-check messages from network nodes. As soon as detection finish, the byzantine nodes could be moved to blacklist and then the process of LA-BFT will return to weak consensus. Once transaction complete, ERA is responsible to update reputations of all nodes and elects leader and candidates for the next round of consensus.

3.2. LA-BFT: Lightweight adaptive Byzantine fault tolerant consensus

In this work, a LA-BFT consensus algorithm is proposed. There are two stages in LA-BFT: weak consensus, byzantine node detection. By the way, reputation calculation method is also elect leader and committee for LA-BFT. The algorithm process is shown in Fig. 2.

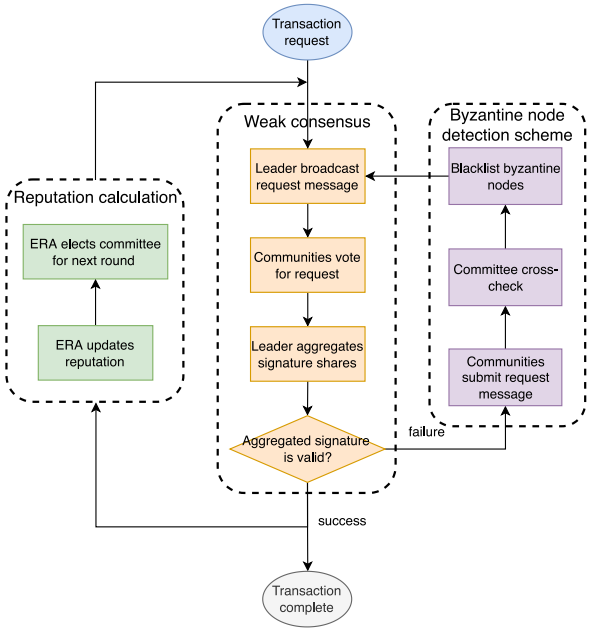


Fig. 2. Simplified byzantine fault tolerant consensus.

3.2.1. Weak consensus of LA-BFT

In the weak consensus, to reduce the communication traffic, threshold signature mechanism is proposed to realize one-to-many communication instead of many-to-many. The algorithmic details of LA-BFT are described next.

In Energy Blockchain network, node N_i holds a public key p and a private key share s_i. With private key share s_i, node N_i can contribute its signature share m^i_{sig} for a message m:

$$m^i_{\text{sig}} = \text{sign}(m, share_i) \tag{1}$$

where m^i_{sig} represents signature share of N_i for message m, and sign() represents signature function.

As for a signature share collection $\{m^i_{\text{sig}}\}_{i\in N}$, when $|N| = k$, where k donates threshold value, those signature shares can be aggregated into a integral aggregated signature m_{sig} by aggregation function aggregation():

$$m_{\text{sig}} = \text{aggregation}(m, \{m^i_{\text{sig}}\}_{i\in N}) \tag{2}$$

All the nodes who hold public key p are able to check the m_{sig} whether correct by verify function verify():

$$\xi = \text{verify}(m, m_{\text{sig}}, p) \tag{3}$$

where $\xi \in \{\text{true}, \text{false}\}$.

The number of communities in the weak consensus should constrained by $N \geq 3f + 1$. Where f is the system tolerable number of byzantine communities. The threshold value k should be $k = 2f + 1$. The process of weak consensus algorithm in detail are as follows:

(1) **Request** phase: The client sends transaction request message $\langle \text{Request}, m, d, t, c\rangle$ to leader that contains the proposal message content m and message digest d. t is the timestamp of Request message generated by the client, and c is the ID of client.
(2) **Prepare** phase: Once leader receives a new Request message from the client, an order transaction number n will be allocated for identifying uniquely Request message. And then leader will broadcast a message $\langle \text{Prepare}, m, d, c, v, n\rangle$ to other communities. v is the consensus view number which indicates a list of nodes status(leader, candidates and communities) in the latest consensus state.

(3) **Prepare-vote** phase: After receiving a new Prepare message, community N_i check the message content m whether is initiated by client c, the message view number v is greater than or equal to local view number v_i of N_i, transaction number n have not submitted in view v. If above sequence of steps being completed correctly, m is signed with private key s_i. Then $\langle \text{Prepare} - \text{vote}, v, n, i, m^i_{sig} \rangle$ is generated and transmitted to the leader. Where m^i_{sig} indicates the signature share for message m and i indicates the identification of N_i.

(4) **PreCommit** phase: After receiving k number of $\text{Prepare} - \text{vote}$ messages, leader aggregates these signature shares according to Eq. (2) to generate a verifiable aggregated signature m_{sig} and verifies whether it is correct. If the verification fails, it indicates that there is byzantine node sending incorrect signature share m^i_{sig} in **Prepare-Vote** phase which results in false in Eq. (3). Therefore, the byzantine node detection scheme is triggered by broadcasting signal message $\langle \text{Change}, v, n \rangle$. If the verification successes, message $\langle \text{PreCommit}, v, n, m_{\text{sig}} \rangle$ will be broadcast to communities.

(5) **PreCommit-vote** phase: After receiving a PreCommit message, community repeatedly checking just like it has done in **Prepare-vote** phase. Something different is that community have to check n that correspond with *Prepare* message that it received. If the verification successes, a message $\langle \text{PreCommit} - \text{vote}, v, n, i, m^i_{\text{sig}} \rangle$ will be transmitted to leader.

(6) **Commit** phase: After receiving k number of $\text{PreCommit} - \text{vote}$ messages, leader aggregates the signature shares just like it has done in **PreCommit** phase. If the verification of aggregated signature is failed, leader broadcast signal message $\langle \text{Change}, v, n \rangle$ to trigger byzantine node detection scheme. Otherwise, a message $\langle \text{Commit}, v, n, m_{\text{sig}} \rangle$ is broadcast to all communities.

(7) **Reply** phase: Once community receives Commit message from leader, it firstly verifies the aggregated signature m_{sig} and check the view number v and transaction number n. If the verification successes, community execute the message content m and append the result s to local blockchain database. Then community sends reply message $\langle \text{Reply}, i, s \rangle$ to client c. When $f+1$ number of *Reply* messages with the same s are arrived, client c can confirm that its request has been commit in blockchain.

In weak consensus, community does not need to communicate with others to obtain the voting information for Request message of all around network. The communities should sign the Request message with private key shares s_i and the leader is responsible for aggregating those signature shares to generate aggregate signature m_{sig}. The aggregated signature m_{sig} is a polymeride of voting information for all communities. Finally, the communities only need to verify the m_{sig} by using public key p to obtain the agreement of entire network for the Request. In this process, there are just $1xN$ messages transmission so that this consensus is of linear communication complexity $O(n)$.

3.2.2. Byzantine nodes detection scheme of LA-BFT

There are two voting phases in weak consensus, namely **Prepare-vote** and **PreCommit-vote**. In voting phases, there are 2 cases need discussing, as shown in Fig. 3.

In normal case, message m is signed by community N_i with private key share s_i to create signature share m^i_{sig}. The leader aggregate the signature shares and verify the validation of aggregated signature. If the verification is successful, the weak consensus process will continue to execute. However, in byzantine case, assume that N_2 is byzantine node. N_2 will send a wrong signature share m^2_{sig} to leader, resulting in aggregated signature invalid. In this case weak consensus process will be blocked.

To recover blocked consensus that cause by byzantine nodes persistently take malicious actions, this work proposes byzantine node detection scheme to prevent them from participating consensus. Suppose that the number of committee members is D. F is the number of byzantine candidates that system can tolerate, there is constraint $D \geq 3F + 1$ as same as constraint in PBFT. The details are as follows:

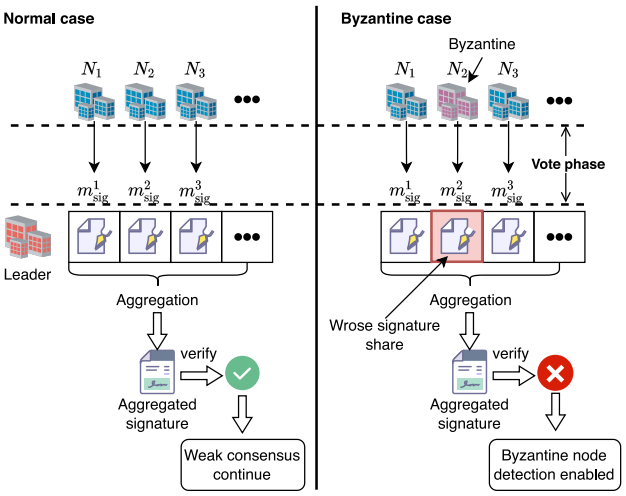


Fig. 3. Two cases in voting phases.

(1) **Submit** phase: After receiving the Change message, community need to broadcast messages $\langle \text{Submit}, v, n, i, m \rangle$ to committee. Where m is the content of Request message that received from leader in the **Prepared-vote** phase to committee.

(2) **Pre-decide** phase: Candidate receives Submit messages from communities and collect statistics on m. If certain m of Submit message from community N_i is mismatched with others, it can be considered that N_i may is byzantine node and be moved to blacklist B. Upon updating blacklist B, candidate broadcasts message $\langle \text{Pre} - \text{Decide}, v, n, i, B \rangle$ to other committee members.

(3) **Decide** phase: After $2F + 1$ number of $\text{Pre} - \text{Decide}$ messages with the same blacklist B arrived, candidate broadcasts message $\langle \text{Decide}, v, n, i, B \rangle$ to other committee members.

(4) **Response phase**: After receiving $2F + 1$ number of Decide messages with the same blacklist B, candidate updates the local blacklist with B and then broadcasts $\langle \text{Response}, i, B \rangle$ to all communities. After receives $F+1$ number of Response messages with the same B, community also updates local blacklist with B and waits for next round of weak consensus.

3.3. Reputation calculation

As the weak consensus is complete, leader and candidates should be selected again for next round of consensus based on the reputation of network nodes. This section presents a reputation calculation method based on multi-dimensional behavioral characteristics.

Let $\Sigma = \{1, 2, 3, \ldots, n\}$ denotes collection of all market participants(including leader, candidates and communities) which meets the constraint of $\Sigma = N + D + L$. Where N donates communities collection, D donates candidates collections and L is the number of leader. By the way, there is only one leader in network, so $|L| = 1$. In each round of consensus, node evaluates the behavior score of each others that it has communicated with. The behavior score of node i received as leader is based on the following rules:

$$E^L_{j \to i, j \in N, i \in L} = \begin{cases} 1, & t_{\text{complete}} \leq t_{\text{timeout}} \\ -1, & j \text{ receive } \textit{Request} \text{ from client} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where t_{complete} indicates the consensus completion time, and t_{timeout} indicates the consensus timeout of consensus. What calls for special

Algorithm 1 Weak consensus phases

Request phase :
Client sends transaction proposal message $\langle \text{Request}, m, d, t, c \rangle$ to the leader.

Prepare phase :
Leader allocate order transaction number n to Request message and broadcast message $\langle \text{Prepare}, m, d, c, v, n \rangle$ to other communities.

Prepare-vote phase :
Communities verify the validity of Prepare message from leader and then transmit message $\langle \text{PreCommit} - \text{vote}, v, n, i, m^i_{\text{sig}} \rangle$ to leader.

PreCommit phase :
After leader collects energy signature shares, a verifiable aggregated signature is generated and a message $\langle \text{PreCommit}, v, n, m_{\text{sig}} \rangle$ is broadcast to all communities.

PreCommit-vote phase :
Communities verify the aggregated signature m_{sig} by public key. Then communities initiate another message $\langle \text{PreCommit} - \text{vote}, v, n, i, m^i_{\text{sig}} \rangle$ to leader.

Commit phase :
Leader aggregates the signature shares from PreCommit − vote again and verifies the legitimacy of aggregated signature. The message $\langle \text{Commit}, v, n, m_{\text{sig}} \rangle$ is then broadcast to all communities.

Reply phase :
Communities verify the aggregated signature from Commit. Then execute content of Request and send execution result message $\langle \text{Reply}, i, s \rangle$ to client c.

attentions is that if the Request message timeout, there are two possibilities: (1) leader is byzantine node and it rejects to broadcast Prepare message; (2) Request message is lost because of network condition is poor. As timeout occur, client will broadcast Request message to communities for next consensus.

The behavior score that node i receives as candidate is based on the following rules:

$$E^C_{j\to i, j\in N, i\in D} = \begin{cases} 1, & i \in \phi \\ -1, & i \notin \phi \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where ϕ donates a collection of $F+1$ number of Response messages.

The behavior score that node i receives as community is based on the following rules:

$$E^N_{j\to i, j\in\{L,D\}, i\in N} = \begin{cases} 1, & m^i_{\text{sig}} \in \{m^i_{\text{sig}}\}_{i\in N, |N|=k} \\ -1, & i \in B \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $\{m^i_{\text{sig}}\}_{i\in N,|N|=k}$ denotes the signature shares used to be aggregated by leader. B indicates the set of blacklisted nodes in this round of byzantine node detection scheme.

Then, the concept of comprehensive evaluation of node's behavior is introduced here which is defined as follows:

$$\overline{E^L_i} = \frac{1}{|N|} \sum_{j\in N} E^L_{j\to i} \tag{7}$$

$$\overline{E^D_i} = \frac{1}{|N|} \sum_{j\in N} E^D_{j\to i} \tag{8}$$

$$\overline{E^N_i} = \frac{1}{|D|} \sum_{j\in D} E^N_{j\to i} + E^N_{j\to i, j\in L} \tag{9}$$

where, $\overline{E^L_i}$, $\overline{E^D_i}$, $\overline{E^N_i}$ respectively represent the comprehensive evaluation received by node i as leader, candidate and community in consensus process. $E_{j\to i}$ indicates the score that node j evaluates to node i.

The behavioral characteristic of node is defined as follows:

$$E^t_i = \omega_1 \overline{E^L_i} + \omega_2 \overline{E^D_i} + \omega_3 \overline{E^N_i} \tag{10}$$

where E^t_i represents the behavioral characteristic of node i. The values of $(\omega_1, \omega_2, \omega_3)$ triples are (1,0,0), (0,1,0) and (0,0,1), depending on what role node i belongs to in the t round of consensus. If it is leader, $(\omega_1, \omega_2, \omega_3) = (1, 0, 0)$; If it is a candidate, $(\omega_1, \omega_2, \omega_3) = (0, 1, 0)$; If it is a community, $(\omega_1, \omega_2, \omega_3) = (0, 0, 1)$.

According to behavioral characteristic of nodes, reputation value will be calculated as follows:

$$\delta^t_i = \begin{cases} 0, & t = 0 \\ \partial \cdot E^t_i \cdot \eta^t_i + \eth \cdot E^t_i \cdot \varphi^t_i + \delta^{(t-1)}_i, & t > 0 \end{cases} \tag{11}$$

$$R^t_i = \frac{1}{1 + e^{-\delta^t_i}} \tag{12}$$

where, R^t_i represents the reputation value of node i in t round of consensus, whose range is [0,1]. δ^t_i is the influence factor, ∂ is the reward coefficient, $\eth$ is the penalty coefficient. In t round of consensus, if the behavioral characteristic of node i is $E^t_i > 0$, then $\eta^t_i = 1$ and $\varphi^t_i = 0$; If $E^t_i < 0$, above values is reversed. Generally reward coefficient ∂ is smaller than penalty coefficient $\eth$ very much. The purpose is to prevent the reputation value of nodes increases too fast and cause the excessive concentration of power. And larger penalty coefficient will intensify punishment to make reputation falling fast when byzantine node take malicious actions at consensus process. This approach encourages nodes to act honestly in consensus.

The penalty coefficient $\eth$ should be influenced by frequency of malicious actions, which is specifically defined as follows:

$$\eth = \text{ך}^{\beta} \tag{13}$$

$$\beta = \frac{\chi_i}{\Delta t_i} \tag{14}$$

where ך donates penalty base value and β donates malicious action frequency of node i. Δt_i represents the total times that node i participates in consensus while χ_i is the number of times that it has acted maliciously.

In order to prevent byzantine nodes from disguising normal action in a few rounds to obtain high reputation, considering that the historical reputation of nodes which can better reflect the reputation degree of nodes. Therefore, the influence factor δ_t should be modified by the bias coefficient θ. The correction formula is as follows:

$$\delta^t_i = \begin{cases} 0, & t = 0 \\ \theta\left(\partial \cdot E^t_i \cdot \delta^t_i + \eth \cdot E^t_i \cdot \varphi^t_i\right) + (1-\theta) \cdot \delta^{(t-1)}_i, & t > 0 \end{cases} \tag{15}$$

where, the range of the bias coefficient θ is (0,1). The smaller θ, the more attention is paid to historical behavior of the node; Otherwise, more attention is paid to the current behavior of the node. In order to prevent nodes from continuously doing maliciously, the reputation threshold is set to a experience value 0.2. When a node continues to act maliciously after its reputation has fallen down to 0.2, its reputation value will be set to 0 as punishment so that it is forbidden to participate in the consensus.

3.4. Security analysis

To verify the security of LA-BFT, we discuss the possible malicious threats of different roles in the consensus process.

3.4.1. Byzantine leader

Theorem 1. *When original leader refuses to broadcast the prepare message, communities can apply for a new leader.*

Proof. If client c receive the consensus result of request message times out, the client c initiates a new request to all communities. The communities will forward the request message to leader and wait for prepare message of this request from leader. If communities receive prepare message times out, it is considered that leader is offline or rejects to broadcast messages. In this case, community sends view-change message to ERA and asks for new leader. □

Theorem 2. *When original leader tampers the content m of request message, communities can apply for a new leader.*

Proof. After receiving the prepare message sent by the leader, the community can verify whether m was created by client c with its public key p through the function $verify()$. If the verification fails, it indicates that the leader tampered with the request message. Community is able to refused to vote on m and initiated view-change to ERA and asks for new leader. □

Theorem 3. *When the request content m in the prepare message played by the original leader is maliciously tampered with an amount less than or equal to f, the weak consensus process continues to execute.*

Proof. After receiving prepare message sent by leader, the community checks whether the message m was created by client c. If the verification fails, it indicates that the leader tampered with the request message. Community is able to refused to vote on m and initiated View-change to ERA and asks for switching leader. Assume that q is the number of tampered prepare messages. If $q <= f$, in the worst case, the leader broadcasts f tampered messages and $2f+1$ correct messages. In voting phase (Prepare-vote phase and PreCommit-vote phase), there are f communities refuse to vote, and $2f+1$ communities agree to vote. Therefore, the number of signature shares received by leader is $2f+1$, achieving the threshold k, so that the aggregated signature is verifiable and the weak consensus process will continue to execute. If $q > f$. In the best case, $f+1$ tempered messages and $2f$ correct messages are broadcast by leader. There are $f+1$ communities refuse to vote, and $2f$ community agree to vote. Therefore, the Leader will receive $2f$ signature shares that they cannot be aggregated because of $2f < k$. The weak consensus process cannot continue to execute. When client find out that request message timeout occurs, it broadcast a new request message to communities just like above Theorem 1. □

3.4.2. Byzantine community

Theorem 4. *When the number of communities that refuse to vote on the request message is less than or equal to f, the weak consensus process continues*

Proof. In the worst case, there are f byzantine communities refuse to vote and $2f+1$ honest communities agree to vote. It means that the leader receives $2f+1$ signature shares and those shares could be aggregated because of $2f+1=k$, k is the threshold of the aggregated signature. Therefore,aggregated signatures is verifiable and the weak consensus process can continue to execute. □

Theorem 5. *When the aggregated signature of request message m is verified fail, the consensus process enables the Byzantine node detection scheme to ensure security.*

Proof. In the worst case, f communities vote on the tampered message, $2f+1$ honest communities agree to vote on the correct message. So that leader receives f wrong signature shares and $2f+1$ correct signature shares. When the leader receives $2f+1$ signature shares and those shares could be aggregated. If one of the top $2f+1$ signature shares which arrived at leader was wrong, the aggregated signature would be verified fail. This situation has been discussed in Section 3.2.2. The consensus process would enable byzantine node detection scheme. After blacklisting byzantine communities, leader broadcast prepare message again to execute weak consensus process. □

3.4.3. Byzantine candidate

Theorem 6. *In order to guarantee the smooth operation of the Byzantine node detection scheme, at most F candidates are tolerated to reject broadcast messages.*

Proof. Suppose that in the Pre-decide phase, in the worst case, there are F byzantine candidates who refuse to broadcast pre-decide messages and $2F+1$ honest candidates do broadcast pre-decide messages. At the start of Decide phase, each candidate will receive $2F+1$ pre-decide messages with the same blacklist B. Since the number of pre-decide messages for the same blacklist B reaches the threshold $2F+1$, honest candidates continue to broadcast decision messages and the Byzantine node detection scheme process will continue. In this case, once the community receives $F+1$ Response messages with the same blacklist B, indicating that it can only come from messages broadcast by honest candidates, the community updates the local blacklist B. □

Theorem 7. *Compromising F candidates to tamper with the blacklist B does not disrupt the smooth running of Byzantine node detection.*

Proof. Suppose that in Pre-decide phase, in the worst case, there are F byzantine candidates broadcasting tempered pre-decide message and $2F+1$ honest candidates broadcasting correct pre-decide messages. At the start of the Decide phase, each candidate will receive $2F+1$ pre-decide messages with the same blacklist B and F pre-decide messages with dissimilar message content. The honest candidate continues to broadcast decide messages and byzantine node detection scheme process will continues to execute, the subsequent procedures are consistent with the above theorem. □

3.5. Summary

This section introduces detailedly LA-BFT consensus algorithm which consist of weak consensus and byzantine nodes detection scheme. The weak consensus introduces threshold signature mechanism so that the leader node can aggregate signature shares into a complete aggregation signature. Any node can verify the correctness of the aggregation signature from leader node based on the public key. so, the nodes only need to communicate with the leader node to reach consensus, rather than communication with each other. Therefore, the LA-BFT algorithm optimizes the communication complexity from $O(n^2)$ to $O(n)$. Moreover, the proposed byzantine nodes detection scheme sets consensus committee to supervise the behavior of consensus nodes. To be specific, consensus committee detect potential byzantine nodes by cross-validation of consensus messages from nodes. The detected byzantine nodes are added into blacklist to prevent them from disturbing consensus results further. The security analysis proves that LA-BFT algorithm has good performance in security.

4. Numerical results

This work uses computer specifications including Intel based processor core i5@2.70 GHz, 16 GB RAM with macOS version 11.7 and OpenJDK version 11.0.9.1 to simulate the distributed energy trading framework.

4.1. Communication cost

The results showed a strong correlation between communication cost and communication complexity of the algorithm. Communication complexity is used to measure the communication difficulty required by the consensus algorithm and is defined as follows:

$$T_{\text{trans}}(N) = \sum_{i \in K} T_i(N) \tag{16}$$

where K donates the number of consensus phases, N donates the scale of consensus network nodes. The value of $T_{\text{trans}}(N)$ depends on the total communication traffic required on the completion of consensus process.

The communication complexity of LA-BFT depends on whether there are byzantine nodes in current network. In order to evaluate LA-BFT more scientifically, we define the average communication complexity:

$$T_{\text{ave}}(N) = (1-\alpha)T_{\text{honest}} + \alpha T_{\text{dishonest}} \tag{17}$$

where α represents the probability of malicious actions occur in the network. On the contrary, $(1-\alpha)$ represents the probability that all participants in the network act honestly.

The communication complexity of PBFT is $O(n^2)$ because each node needs to broadcast messages and each round of consensus requires $2n^2 - n + 1$ times of communication. In the case of no byzantine nodes in network, the weak consensus proposed in this work only needs communities to interact with the leader, wherein each round of consensus just needs $6n - 1$ times of communication. So the communication complexity of weak consensus is $O(n)$. In the case of byzantine node detection scheme been enabled, the process requires $2D(n-1)$ times of communication and hence the communication complexity is $O(Dn)$, where $D \ll n$.

Fig. 4 illustrates a comparison of communication traffic between LA-BFT, PBFT and Scalable hierarchical Byzantine fault tolerance (SHBFT). With the increase of the number of consensus nodes, the communication times of different consensus algorithms also grow to different degrees. Among them, PBFT requires much higher number of communications than SHBFT and LA-BFT under the same conditions. SHBFT performs partition optimization based on PBFT, Narrows the consensus range, and reduces the number of communications. And as the number of nodes continues to increase, the gap between SHBFT and LA-BFT becomes larger and larger. What is more, with the increase of α, the communication number of LA-BFT will grow accordingly, but it still shows a linear growth trend. And even if it is 0.3, the communication times of LA-BFT are only 7.66% of PBFT and 30.83% of SHBFT.

Communication cost, in the context of blockchain networks, refers to the extent of network bandwidth usage during the process of consensus. Less bandwidth usage indicates better performance of the consensus algorithm. The experiment set up the client to concurrently send different transaction requests, and records the bandwidth usage with different number of nodes. Result is shown in Fig. 5.

As the number of concurrent transactions and nodes increase, the bandwidth usage of different consensus algorithms also increases. Among them, PBFT has the largest increase in bandwidth usage, followed by SHBFT and finally LA-BFT. Especially, when the number of concurrent transactions is 2000 and the number of nodes is 50, the bandwidth requirement of LA-BFT is only 5.55% of PBFT and 9.76% of SHBFT. It can be seen from Fig. 5 that, in the same case, as the value of increases, the bandwidth occupation of LA-BFT also increases. When the number of concurrent transactions is 2000 and the number of nodes is 50, the bandwidth requirement of LA-BFT ($\alpha = 0.3$) is 321.22% of LA-BFT ($\alpha = 0.1$) and 137.29% of LA-BFT ($\alpha = 0.2$). The reason is that the higher the evil probability of Byzantine nodes, the stronger consensus is triggered more frequently by LA-BFT, which increases the communication cost. However, compared with PBFT and SHBFT, LA-BFT still has obvious advantages. The bandwidth requirement of LA-BFT ($\alpha = 0.3$) is only 19.99% of PBFT and 35.14% of SHBFT.

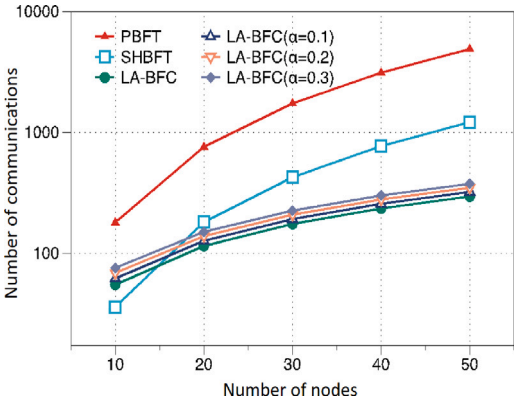


Fig. 4. Comparison of communication traffic.

All in all, LA-BFT can greatly save bandwidth and reduce the risk of transaction timeout due to network congestion.

4.2. Latency

Consensus latency refers to the transaction delay, which represents the interval between client sending a transaction request to blockchain network and the client confirming completion of consensus process.

The experiment set up same concurrent transactions and different number of nodes to record average transaction delay. After multiple tests, the average consensus delay was calculated, and the results are shown in Fig. 6.

From Fig. 6, we can see that with the increase of the number of nodes, the average transaction delay of different algorithms will increase to varying degrees. However, LA-BFT is less affected by the number of concurrent transactions, whereas PBFT and SHBFT are substantially affected. In all concurrent cases, initially, under a certain number of nodes, PBFT and SHBFT exhibit a lower average transaction delay than LA-BFT. Once the number of nodes exceeds the certain threshold, the average transaction delay of PBFT and SHBFT is higher than LA-BFT. As the number of nodes continues to increase, not only the delay gap between PBFT and LA-BFT but also the delay gap between SHBFT and LA-BFT progressively widen.

For example, when the number of nodes is 10, the average consensus delay of PBFT is the lowest, which is 83.58% of SHBFT and 37.59% of LA-BFT. However, when the number of nodes increases to 20, the average consensus delay of PBFT increases rapidly and exceeds that of SHBFT and LA-BFT. When the number of nodes continues to increase to 30, the average consensus delay of SHBFT also exceeds that of LA-BFT, and LA-BFT is 59.83% of it. At this time, the advantage of LA-BFT gradually expands. Especially, when the number of nodes increases to 50, the average consensus delay of LA-BFT is only 16.19% of PBFT and 33.58% of SHBFT.

Compared with LA-BFT, LA-BFT ($\alpha = 0.3$) has higher delay. Because of byzantine node detection scheme is enabled to blacklist byzantine nodes when those them act maliciously in curtain rounds. The detection process costs more four phases of communication than weak consensus of LA-BFT which results in a sudden increase in transaction delay compared with normal situation. However, it is still better than PBFT and SHBFT. For example, when the probability of Byzantine node evil is

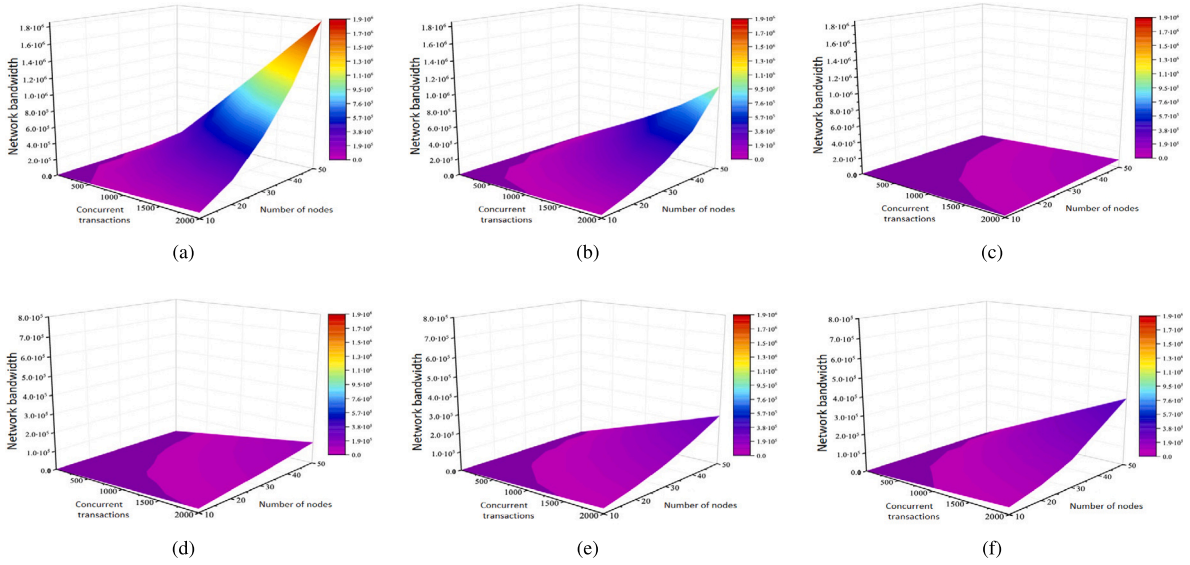


Fig. 5. Bandwidth usage of consensus: (a) PBFT; (b) SHBFT; (c) LA-BFT; (d) LA-BFT ($\alpha = 0.1$); (e) LA-BFT ($\alpha = 0.2$); (e) LA-BFT ($\alpha = 0.3$).

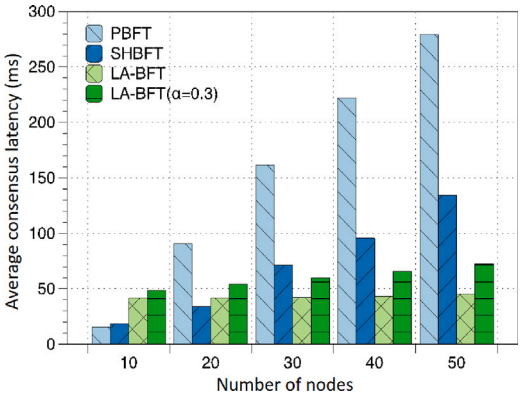
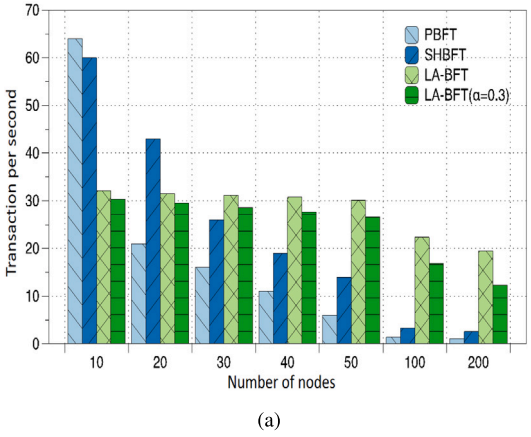


Fig. 6. Comparison of average consensus delay among different consensus algorithms.



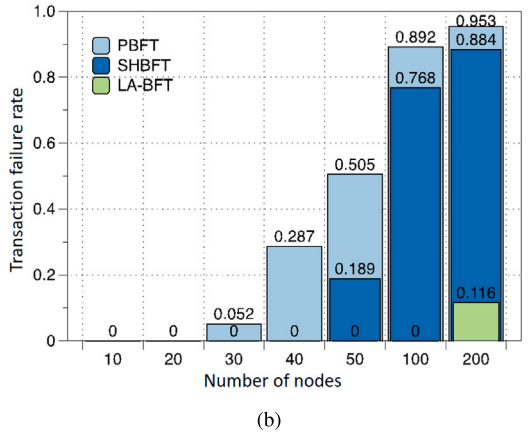


Fig. 7. Comparison of throughput and transaction failure rate: (a) Throughput; (b) Transaction failure rate.

0.3, the average consensus delay of LA-BFT ($\alpha = 0.3$) is 60.71% higher than that of LA-BFT, but 25.89% of PBFT and 53.73% of SHBFT.

In conclusion, PBFT and SHBFT exhibit superior concurrency capacity under small-scale scenarios. Whereas LA-BFT outperforms PBFT and SHBFT significantly in larger-scale scenarios.

4.3. Throughput

Throughput refers to the number of transactions completed per unit time and is generally expressed as tps. In the experiment, client was set up to randomly persistently send 2000 transaction requests when the number of nodes is 10, 20, 30, 40, 50, 100 and 200. Fig. 7(a) shows comparison of different consensus algorithms for throughput.

As the number of nodes increases, the TPS of the consensus algorithm decreases. When number of nodes is 10, PBFT gets maximum throughput compared with both SHBFT and LA-BFT, SHBFT is second only to PBFT, and LA-BFT has the smallest TPS. The reason is that when the number of nodes is small, the bandwidth occupation of PBFT and SHBFT is low and the communication between nodes is smooth, and the delay is mainly affected by the number of communication stages in consensus. PBFT has 5 phases of communication, SHBFT has 9 phases of communication, and LA-BFT has 7 phases of communication, in which case PBFT has the lowest communication delay and therefore the highest TPS. And because LA-BFT needs to aggregate the threshold signature shares of nodes in the consensus process, the aggregation time affects the overall communication delay, so the TPS is the lowest.

However, as the number of nodes continues to increase, the bandwidth requirement of PBFT and SHBFT rises rapidly, which easily

causes network congestion and increases the communication delay, resulting in a rapid reduction of the TPS of PBFT and SHBFT. However, since LA-BFT uses one-to-many communication mode, the increase of the number of nodes has little impact on the bandwidth occupation, so the fluctuation of its TPS does not change significantly. When the number of nodes reaches 200, the TPS of LA-BFT is 17.71 times that of PBFT and 7.28 times that of SHBFT. Meanwhile, the TPS of LA-BFT ($\alpha = 0.3$) is 11.22 times that of PBFT and 4.81 times that of SHBFT. As Fig. 5 shows that the bandwidth usage of LA-BFT is less than PBFT and SHBFT. It can be seen that, compared with PBFT and SHBFT, LA-BFT is more suitable for application in large-scale electric energy trading scenarios.

The experiment also compares the transaction failure rate of LA-BFT, PBFT and SHBFT, as Fig. 7(b) shows. In the experiment, the transaction timeout time is set to 90 s, the client concurrent 2000 transactions, and the transaction failure rate of different consensus algorithms is also tested when the number of nodes is $10, 20, 30, 40, 50, 100,$ and 200 respectively. When the number of PBFT nodes is 30, the transaction failure rate starts to increase. SHBFT starts to increase when the number of nodes is 50. In contrast, LA-BFT has transaction invalidation only when the number of nodes is 200. When the number of nodes is 200, the failure rate of LA-BFT is only 12.17% of PBFT and 13.12% of SHBFT.

4.4. Node reputation

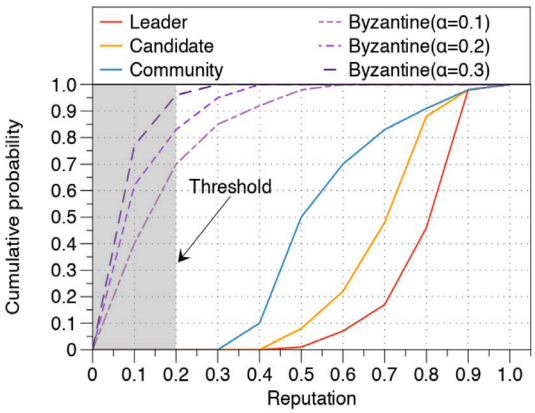
This work simulates the changes of the reputation value of 50 nodes in 50 rounds of consensus, among which there are 13 byzantine nodes, 9 candidates, 27 communities and 1 leader with parameters $\partial = 1$ and $\daleth = 10$. The system randomly selects candidate and leader at the beginning. After first update of reputation, the top 30% is selected as candidate, and the node with the highest reputation is leader. The initialized reputation of nodes is 0.5. The experiment recorded the cumulative distribution function (CDF) of reputation in 50 rounds of consensus. The comparison of CDF of different nodes is shown in Fig. 8(a).

The byzantine nodes generally get a lower reputation value than communities. More than 70 percent of byzantine nodes get reputation of 0.2 or less while 90 percent of communities has reputation of 0.4 or more. That is why the reputation threshold is set by 0.2 in this work. With the increase of α, byzantine nodes' reputation becomes lower and lower. At the same time, more than 50 percent of candidates and leader have reputation value above 0.8 which is proved that byzantine nodes are unlikely elected as candidates or leader.

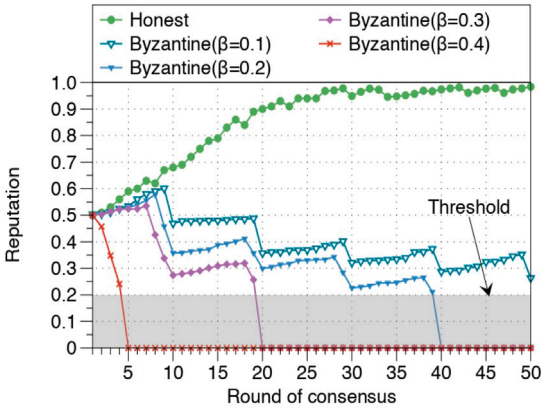
We compare the reputation changes of a certain honest node and three byzantine nodes with different evil frequency β. The results is shown in Fig. 8(b). With the advance of consensus round, the reputation of honest node rises steadily. For byzantine nodes, every malicious actions they take diminishes their reputation. And the greater the number of β, the greater the penalty for reputation. Because of penalty coefficient $\eth$ is related to the node's evil frequency β, the larger β is, the larger $\eth$ will be, and the faster reputation will decline. The smaller reputation value, the more consensus round s the node needs to spend to improve its reputation. When reputation is lower than the reputation threshold value 0.2, it will be directly update to 0 and participation in consensus will be prohibited.

The results show that the consensus mechanism designed in this work can effectively reduce the reputation value of byzantine nodes and prevent byzantine nodes from becoming leader or candidates, which ensures the security of LA-BFT.

The limitations of this work are summarized as follows: (i) compared with PBFT and SHBFT algorithm, LA-BFT algorithm has higher latency and lower throughput in small-scale scenarios. (ii) when byzantine nodes take malicious actions in consensus procedure, the efficiency of LA-BFT algorithm is reduced. (iii) the proposed reputation calculation method requires several consensus rounds to filter out byzantine nodes.



(a)



(b)

Fig. 8. Reputation of nodes: (a) Cumulative distribution functions of nodes; (b) Comparison of reputations between different nodes.

5. Conclusions

In this work, a consensus algorithm named LA-BFT is proposed which consists of weak consensus process, byzantine node detection process. In scenarios involving normal conditions and the detection of Byzantine nodes, our algorithm exhibits time complexities of $O(n)$ and $O(n^2)$ respectively. In addition, a new reputation calculation method is presented to elect leader and candidates for consensus procedure, which will bring in additional communication overhead. With the reputation calculation method, malicious nodes can be effectively identified, ensuring the security of the consensus process. Simulation results show that the performance of the proposed method is significantly better than that of PBFT and SHBFT in large-scale distributed trading scenarios. Specifically, when the number of nodes is 50, under normal scenarios, the communication overhead of LA-BFT is notably lower, constituting a mere 6.02% of PBFT and 24.26% of SHBFT.

In future works: Although the proposed LA-BFT consensus algorithm reduces the communication complexity to linear complexity, the efficiency will still be reduced when handling with byzantine cases. Therefore, future work will focus on how to improve the efficiency of byzantine node detection processing in LA-BFT.

CRediT authorship contribution statement

Jin Ye: Supervision, Project administration, Conceptualization, Formal analysis, Funding acquisition, Resources. **Huilin Hu:** Writing –

review & editing, Conceptualization, Formal analysis, Investigation, Methodology. **Jiahua Liang:** Writing – original draft, Data curation. **Linfei Yin:** Software, Validation. **Jiawen Kang:** Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

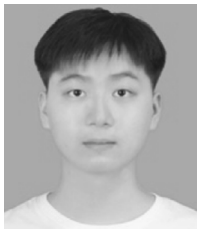
This work was supported by the National Natural Science Foundation of China under Grant U22A2021, U23A20313 and 52107081.

References

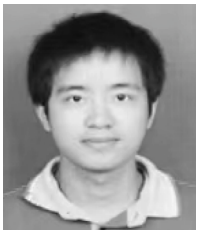
[1] L. Yin, Y. Qiu, Long-term price guidance mechanism of flexible energy service providers based on stochastic differential methods, Energy 238 (2022) 121818.
[2] D. Han, C. Zhang, J. Ping, Z. Yan, Smart contract architecture for decentralized energy trading and management based on blockchains, Energy 199 (2020) 117417.
[3] Q. Yang, H. Wang, T. Wang, S. Zhang, X. Wu, H. Wang, Blockchain-based decentralized energy management platform for residential distributed energy resources in a virtual power plant, Appl. Energy 294 (2021) 117026.
[4] A. Umar, D. Kumar, T. Ghose, Blockchain-based decentralized energy intra-trading with battery storage flexibility in a community microgrid system, Appl. Energy 322 (2022) 119544.
[5] W. Tushar, T.K. Saha, C. Yuen, D. Smith, H.V. Poor, Peer-to-peer trading in electricity networks: An overview, IEEE Trans. Smart Grid 11 (4) (2020) 3185–3200.
[6] M. Dong, L. Li, Y. Nie, D. Song, J. Yang, Stability analysis of a novel distributed secondary control considering communication delay in DC microgrids, IEEE Trans. Smart Grid 10 (6) (2019) 6690–6700.
[7] L. Yin, Y. Qiu, Neural network dynamic differential control for long-term price guidance mechanism of flexible energy service providers, Energy 255 (2022) 124558.
[8] M. Mehdinejad, H. Shayanfar, B. Mohammadi-Ivatloo, Decentralized blockchain-based peer-to-peer energy-backed token trading for active prosumers, Energy 244 (2022) 122713.
[9] M. Nykyri, T.J. Kärkkäinen, S. Levikari, S. Honkapuro, S. Annala, P. Silventoinen, Blockchain-based balance settlement ledger for energy communities in open electricity markets, Energy 253 (2022) 124180.
[10] S. Junjia, D. Ziming, H. Junjie, H. Qiru, W. Fei, P2P smart power trading contract based on blockchain technology, Power Syst. Technol. 45 (10) (2021) 3830–3839.
[11] T. Li, D. Li, M. Wang, Blockchain-based fair and decentralized data trading model, Comput. J. 65 (8) (2021) 2133–2145.
[12] X. Zhang, M. Dong, J. Ou, A distributed cooperative control strategy based on consensus algorithm in dc microgrid, in: 2018 13th IEEE Conference on Industrial Electronics and Applications, ICIEA, IEEE, 2018, pp. 243–248.
[13] B. Wang, L. Xu, J. Wang, A privacy-preserving trading strategy for blockchain-based P2P electricity transactions, Appl. Energy 335 (2023) 120664.
[14] A. Esmat, M. de Vos, Y. Ghiassi-Farrokhfal, P. Palensky, D. Epema, A novel decentralized platform for peer-to-peer energy trading market with blockchain technology, Appl. Energy 282 (2021) 116123.
[15] D. Zhaoyang, L. Fengji, L. Gaoqi, Blockchain: A secure, decentralized, trusted cyber infrastructure solution for future energy systems, J. Mod. Power Syst. Clean Energy 6 (5) (2018) 958–967.
[16] Y. Liu, H. Yu, W. Wang, S. Zou, D. Liu, D. Gong, Z. Li, A robust blockchain-based distribution master for distributing root zone data in DNS, Comput. J. 65 (11) (2022) 2880–2893.
[17] S. Saxena, H. Farag, A. Brookson, H. Turesson, H. Kim, Design and field implementation of blockchain based renewable energy trading in residential communities, in: 2019 2nd International Conference on Smart Grid and Renewable Energy, SGRE, IEEE, 2019, pp. 1–6.
[18] L. Duan, W. Xu, W. Ni, W. Wang, BSAF: A blockchain-based secure access framework with privacy protection for cloud-device service collaborations, J. Syst. Archit. 140 (2023) 102897.
[19] A. Esmat, M. de Vos, Y. Ghiassi-Farrokhfal, P. Palensky, D. Epema, A novel decentralized platform for peer-to-peer energy trading market with blockchain technology, Appl. Energy 282 (2021) 116123.
[20] X. Fu, H. Wang, P. Shi, X. Zhang, Teegraph: A blockchain consensus algorithm based on TEE and DAG for data sharing in IoT, J. Syst. Archit. 122 (2022) 102344.
[21] S. Liu, F. Chen, L. Shen, Y. Hu, Y. Ding, A high-performance local energy trading cyber-physical system based on blockchain technology, in: IOP Conference Series: Earth and Environmental Science, vol. 227, (no. 3) IOP Publishing, 2019, 032009.
[22] A. Singh, G. Kumar, R. Saha, M. Conti, M. Alazab, R. Thomas, A survey and taxonomy of consensus protocols for blockchains, J. Syst. Archit. 127 (2022) 102503.
[23] J. Yang, A. Paudel, H.B. Gooi, H.D. Nguyen, A proof-of-stake public blockchain based pricing scheme for peer-to-peer energy trading, Appl. Energy 298 (2021) 117154.
[24] J. Peipei, W. Qian, C. Yanjiao, L. Qi, S. Chao, Securing guarantee of the blockchain network: Attacks and countermeasures, J. Commun. 42 (1) (2021) 151–162.
[25] C. Dang, J. Zhang, C.-P. Kwong, L. Li, Demand side load management for big industrial energy users under blockchain-based peer-to-peer electricity market, IEEE Trans. Smart Grid 10 (6) (2019) 6426–6435.
[26] W. Dewen, W. Lixin, Multi-energy interaction subject consensus mechanism based on practical Byzantine fault-tolerant algorithm, Autom. Electr. Power Syst. 43 (2019) 9.
[27] M. Perc, The Matthew effect in empirical data, J. R. Soc. Interface 11 (98) (2014) 20140378.
[28] Y. Cheng, X. Hu, J. Zhang, An improved scheme of proof-of-stake consensus mechanism, in: 2019 4th International Conference on Mechanical, Control and Computer Engineering, ICMCCE, IEEE, 2019, pp. 826–8263.
[29] M. Zulfiqar, M. Kamran, M. Rasheed, A blockchain-enabled trust aware energy trading framework using games theory and multi-agent system in smat grid, Energy 255 (2022) 124450.
[30] J. Yang, J. Dai, H.B. Gooi, H.D. Nguyen, P. Wang, Hierarchical blockchain design for distributed control and energy trading within microgrids, IEEE Trans. Smart Grid 13 (4) (2022) 3133–3144.
[31] X. Luo, K. Xue, J. Xu, Q. Sun, Y. Zhang, Blockchain based secure data aggregation and distributed power dispatching for microgrids, IEEE Trans. Smart Grid 12 (6) (2021) 5268–5279.
[32] S. Chen, Z. Shen, L. Zhang, Z. Yan, C. Li, N. Zhang, J. Wu, A trusted energy trading framework by marrying blockchain and optimization, Adv. Appl. Energy 2 (2021) 100029.
[33] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, F. Han, An architecture and performance evaluation of blockchain-based peer-to-peer energy trading, IEEE Trans. Smart Grid 12 (4) (2021) 3364–3378.
[34] J. Ping, Z. Yan, S. Chen, A privacy-preserving blockchain-based method to optimize energy trading, IEEE Trans. Smart Grid 14 (2) (2022) 1148–1157.
[35] T. Wang, J. Guo, S. Ai, J. Cao, RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration, Appl. Energy 295 (2021) 117056.

Jin Ye received Ph.D. degree from Central South University, Changsha, china in 2008. Since 2012, she has been a professor and master supervisor of the School of Computer and Electronic Information, Guangxi University. Her research interests include network protocol optimization, wireless network design, protocol attack and defense.

Huilin Hu received bachelor degree from Southwest University of Finance and Economics, Chengdu, china in 2019, and has been studying in the School of Computer and Electronic Information, Guangxi University since 2023, focusing on privacy computing and security protection.



Jiahua Liang received the master degree from Guangxi University, Nanning, china in 2023. His main research direction is blockchain systems, and he is committed to researching consensus algorithms that resist Byzantine attacks.



Linfei Yin received Ph.D. degree from South China University of Technology, Guangzhou, china in 2018. His main research direction is Automatic power generation control, and he is committed to load forecasting and new energy output forecasting.

Jiawen Kang received the Ph.D. degree from the Guangdong University of Technology, Guangzhou, China in 2018. From 2018 to 2021, he was a Postdoc with Nanyang Technological University, Singapore. He is currently a Professor with the Guangdong University of Technology. His research interests include blockchain, security, and privacy protection in wireless communications and networking.