Dynamic reputation–based consensus mechanism: Real-time transactions for energy blockchain

Wenjun Cai\textsuperscript{1,2}, Wei Jiang\textsuperscript{3}, Ke Xie\textsuperscript{4}, Yan Zhu\textsuperscript{1,2}, Yingli Liu\textsuperscript{1,2} and Tao Shen\textsuperscript{1,2}

Abstract
The energy blockchain is a distributed Internet protocol for energy transactions between nodes in power systems. The consensus algorithm is the core component of the energy blockchain and has an essential impact on its application. At present, in the implementation of the energy blockchain, there are problems such as low transaction throughput (transactions per second) and high latency, which cannot meet the application requirements of real-time processing transactions in the energy field. To this end, according to the analysis of conventional blockchain consensus algorithm and traditional practical Byzantine fault tolerance algorithm, a dynamic-reputation practical Byzantine fault tolerance algorithm for the energy blockchain is proposed. The dynamic-reputation practical Byzantine fault tolerance algorithm adopts a credit-based consortium node consensus election method. The monitoring node divides the remaining nodes into two types of nodes according to the reputation value: the consensus node and the secondary node, which, respectively, participate in different stages of the block generation process, and dynamically update the consensus nodes with low reputation ratings. By constructing the experimental platform simulation, the test results verify the effectiveness of the dynamic-reputation practical Byzantine fault tolerance algorithm. Compared with the algorithm of the fabric platform, the dynamic-reputation practical Byzantine fault tolerance algorithm improves the transaction processing speed and is suitable for the blockchain application in the energy field.

Keywords
Energy blockchain, practical Byzantine fault tolerance, consensus algorithm, reputation value

Date received: 15 October 2019; accepted: 22 January 2020

Handling Editor: Bo Rong

Introduction
In the traditional energy market, when trading between the main entities, the centralized model is adopted: the generators and users upload the data to the central processing system and match the sending and receiving parties through centralized matching or optimization.\textsuperscript{1} However, with the promotion of distributed renewable energy, the trading mode of energy Internet tends to the trend of diversification of subjects, diversification of commodities, decentralization of decision-making, transparency of information, and timeliness of trading.\textsuperscript{2–4} The method of centralized decision-making will increase the operating cost of the trading center as well as the trading time.\textsuperscript{5} If the trading center has a black
box operation or is subject to external hacking, the security of the trade and the privacy of the participating entities cannot be guaranteed. Therefore, the existing centralized energy trading model is challenging to meet the development requirements of the energy Internet. New technology is needed to transform the current energy trading model and realizes the transformation of the distributed trading model instead of the centralized trading model.

Blockchain technology is a decentralized emerging technology. It consists of in-tamperable block data and executable smart contracts code. It has the characteristics of distributed decision-making, coordinated autonomy, high security, and tamper-proof transparency. Due to these characteristics of the blockchain, industry and academia have gradually realized that blockchain technology is compatible with the operation mode, topology, and security of the energy Internet, which can effectively improve the efficiency of transaction processing in energy Internet. It can be said that the technical characteristics of the blockchain itself have the inherent advantages of reconstructing the energy system.

At present, preliminary achievements have been made in the research and application of blockchain technology in the energy field. From the recent published literature of blockchain technology in the field of energy, we can see that the application research of blockchain technology in energy can be divided into four categories. The first category is to analyze the impact of the application of blockchain technology in the energy market. The second category is the capability analysis of applying feasibility and trading mechanisms. The third category is smart contracts that design transaction processes and transactions. The fourth category focuses on the analysis of energy loss attribution during the energy blockchain transaction. From these application studies, the typical energy blockchain trading scenario can be derived, as shown in Figure 1.

$n$ users are built into a blockchain platform, distributed energy and users send their own trading needs to the platform through blockchain smart contracts and then match the trading needs through relevant mechanisms, and finally confirm the transactions through the

![Figure 1. Typical energy blockchain trading scenario.](image-url)
blockchain. Various consensus algorithm generation blocks are stored at each node, and the smart meter and controller (SMC) receives the transaction smart contract that has been processed and confirmed. However, in the consensus section of this energy blockchain trading scenario, the performance of the energy blockchain will be different when different consensus algorithms are chosen. J Wang et al.\(^\text{19}\) proposed a dynamic quotation method to conduct electricity trading through the combination of blockchain and continuous double auction mechanism, but they chose the proof-of-work (PoW) algorithm. Since PoW requires each consensus node of the entire network to calculate a mathematical problem through a large number of resources, there is a problem that energy consumption is severe, and it is challenging to settle energy transactions in real time. Q He et al.\(^\text{21}\) used the traditional practical Byzantine fault tolerance (PBFT) algorithm to design and realize the charging pile transaction. Because in each round of consensus process, all nodes need to participate in the consensus, there is a problem of low consensus efficiency. Therefore, the core of the performance of the energy blockchain lies in the design of consensus algorithms. The consensus algorithm is a method to ensure the voting, and data records of different nodes are consistent in a distributed network environment.\(^\text{25}\) The existing blockchain consensus algorithms include PoW, proof of stake (PoS), and PBFT protocol with Hyperledger Fabric as an example, and so on.\(^\text{26}\) Among them, although PoW is recognized as the best solution to the problem of Byzantine fault tolerance, it can only be used in the public chain and has a high energy consumption and high cost of use, so it cannot meet the needs of the energy Internet.\(^\text{27}\) At the same time, the PoS algorithm determines the right to write blocks based on the wealth possessed by the nodes. Therefore, there are fairness and security issues, and PoS cannot meet the needs of the energy Internet. Although the PBFT algorithm of the consortium blockchain is not as energy-intensive as PoW and guarantees security, its primary node needs to be generated by the polling mechanism. Moreover, the unnecessary overhead of communication in the consensus process is not fully applicable to the energy field with larger nodes.

In summary, in the energy blockchain consensus link, the node consensus efficiency is low, and it is difficult to settle the transaction in real-time, this article analyzes the blockchain performance problem of the traditional PBFT algorithm and adopts the system architecture of the consortium blockchain with audit mechanism to propose a suitable Internet for energy. The dynamic-reputation practical Byzantine fault tolerance (DPBFT) algorithm divides the nodes in the blockchain network according to their credit values to form a set of nodes with different operational rights. A few nodes participate in the consensus process, which reduces the communication interaction cost and computational pressure of legitimacy confirmation to improve the consensus efficiency. It has the advantages of durable consistency, high throughput, low latency, and low central processing unit (CPU) usage. Finally, the effectiveness of the DPBFT algorithm is verified by building an experimental platform simulation.

### PBFT algorithm

Most of the traditional consensus algorithms rarely consider the Byzantine fault tolerance problem, and the PBFT algorithm solves the problem that the original Byzantine fault-tolerant algorithm is not efficient, and reduces the algorithm complexity from exponential to polynomial, making the PBFT algorithm feasible in practical system applications. Simultaneously, PoW and PoS algorithms are widely used in the virtual currency application of the public blockchain, and it is difficult to use them in real-world applications. Therefore, most applications still use the PBFT algorithm. For example, the blockchain system Hyperledger Fabric 0.6 is implemented by the PBFT algorithm.

PBFT is an algorithm based on state machine replica, that is, the state machine performs the same result on any replica node under the same state and input parameters. In the meantime, all replica nodes must be executed from the same state.\(^\text{28}\) The algorithm provides fault-tolerant guarantee that the failed node does not exceed \((n - 1)/3\) on the premise of ensuring system activity and security. Namely, if more than two-thirds of the nodes are healthy, the whole system can usually work.\(^\text{29}\)

The node of the algorithm has two roles: a primary node and lots of replica nodes. In a round of consensus, only the primary node is responsible for sorting the transaction requests received during the period and then sending them to all the replica nodes by a sequence number. Finally, the verified transactions will be packaged in the blockchain. All consensus nodes have different names from 0 to \(|R| - 1\), and the primary node selection formula is

\[
p = v \mod (|R|)
\]

where \(p\) is the quantity selected as the primary node; \(v\) is the view number, which gradually increments from zero; and \(|R|\) is the total number of all consensus nodes.

In the case that the primary node is normal, a complete node consensus process requires a three-phase communication process, as shown in Figure 2. The specific steps are as follows:

1. “Pre-Prepare” phase: the primary node generates a prepared message \(< PRE\_PREPARE, v, n, d>, m >\) according to the transaction
request and sends it to each replica node (where \( n \) is the serial number assigned by the primary node to the request, \( m \) is the client’s request message, and \( d \) is \( m \)'s digest). Each replica node starts to verify the correctness of the message and then the primary node enters the “Prepare” phase.

2. “Prepare” phase: after each replica node receives the pre-preparation message, each replica node generates a preparation message \(<PREPARE, v, n, d, i>\) and sends it to other nodes (including the primary node). And entering the “Commit” stage after each node receives the preparation message and verifies whether \((m, v, n, i)\) it is true.

3. “Commit” phase: each node generates a commit message \(<COMMIT, v, n, D(m), i>\) and sends it to other nodes to verify the vote. After a node receives the confirmation information of \(2f + 1\) (\(f\) represents the maximum number of malicious nodes that can be tolerated) “Commit” from other nodes (including itself), the block will be written into the end of the blockchain.

As shown in Figure 2, “Primary Node 1” is the current primary node. “Replica Node 1,” “Replica Node 2,” and “Replica Node n” are \(n - 2\) honest and delay-free replica nodes, and “Replica Node 4” has a passive error, even if “Replica Node 4” has a Byzantine error, according to the same voting process agreement, it can be executed normally. This shows that in a complete consensus process of PBFT,

1. The primary node needs to poll all participating nodes randomly, and all nodes need to participate in consensus operation.
2. It is necessary to go through the communication processes with complexity \(O(n^2)\), which is high in complexity. The total number of broadcast executions of the communication process is

\[
f_{PBFT}(n) = 2n(n - 1) \tag{2}
\]

In the energy blockchain, the number of nodes and performance requirements are increasing. All nodes participate in the selection and consensus operations, and the consensus performance of PBFT will drop dramatically. If the communication takes 5 ms at a time in a communication environment, it takes time \(T_t\) to complete a block confirmation

\[
T_t = \frac{n(n - 1)}{6000 \text{ min}} \tag{3}
\]

When the number of nodes reaches 35, it takes up to 2 min to generate a block. Therefore, in the case of many nodes, the traditional PBFT algorithm cannot be fully applied to the energy field of real-time trading. In response to these problems, this article proposes a consensus election method based on the credit value to improve the PBFT algorithm.

**PBFT algorithm optimization**

**Consensus model**

The blockchain consensus model presented in this article is shown in Figure 3.

At present, the blockchain is divided into public blockchain, consortium blockchain, and private blockchain. The public blockchain is open to everyone and anyone can participate in consensus, which is completely “decentralized,” but there are problems with low efficiency of transactions and difficult to guarantee...
user privacy. The consortium blockchain and the private blockchain are broadly defined private blockchain, in which only a few nodes have read/write rights to the blockchain; however, the private blockchain is only open to independent individuals or entities and is not much different from other distributed storage solutions. In the energy field, comprehensively considering privacy, communication, and storage efficiency, the consortium blockchain is more suitable for the energy market environment. Therefore, this model uses the consortium blockchain architecture, assuming that the total node set is $R$ and the node number in the set is $R_i (i \in [0, |R| - 1])$, the DPBFT consensus algorithm divides the $|R|$ nodes into three types of node sets: $k$ consensus nodes $S_{CN} = \{S_{CN1}, S_{CN2}, \ldots, S_{CNk}\}$, $m$ secondary nodes $S_{SN} = \{S_{SN1}, S_{SN2}, \ldots, S_{SNm}\}$ ($0 < m < k < |R|$), and a monitoring node $C$, which, respectively, participate in different stages of the block generation process and dynamically update the consensus nodes with low reputation ratings. The monitoring node $C$ periodically monitors the status information of the scanning consensus node $S_{CN}$ and the secondary node $S_{SN}$. In the $r$-round consensus process, the primary node $S_{CNp}$ ($\forall S_{CNp} \in S_{CN}$) obtains the transaction data creation block from the cache database and propagates the storage information to the consensus node $S_{CN}$ and the secondary node $S_{SN}$. At the beginning of the $r + 1$ round, the monitoring node $C$ calculates and updates the reputation value $f_{RV}$ of each node according to the contribution of each node of the $r$ round to the system. Finally, the operation authority of each node in the $r + 1$ round is determined by the reputation value.

In the consensus model, only the consensus nodes are allowed to participate in the voting verification operation, so that the consensus and the high-power consumption operation can be executed separately, avoiding all nodes participating in consensus operations. Besides, the consensus process confirms the legitimacy of the block through the primary node, reduces the communication complexity of the original PBFT, and shortens the time required for the consensus cycle, thereby improving the consensus efficiency.

**DPBFT algorithm concept**

The DPBFT algorithm is defined as follows.

**Monitoring node.** In the DPBFT algorithm, the monitoring node is the hub of each consensus node and the secondary node. In the consensus process, the main functions of monitoring node are as follows:

1. Accept the heartbeat information sent by the heartbeat thread of each node and process it.
2. Registered node: the newly registered node sends the node information to the monitoring node in the blockchain. The monitoring node verifies whether the node is legal. After the
successful verification, the monitoring node calculates the reputation value for the new node, assigns the corresponding authority, and broadcasts it to the whole network.

3. Select the primary node: obtain a parameter for calculating a reputation value, calculate a reputation value of each node, sort the node type dynamically according to the reputation value, select a primary node that satisfies the condition in the consensus node, and place the primary node into the static variable.

4. Open a new round of consensus: after verifying that the status of each node is consistent, notify the primary node to open a new round.

5. Restart of the node: first, it verifies whether the node is in the monitoring, determines whether the node is the primary node of the previous round, clears the static variable of the primary node, and then decides whether the node needs data synchronization, and finally modifies the status of node and calculates its credit value division authority.

6. Verify that the synchronization is complete.

**Consensus node.** The consensus node (CN) is the first 60% node selected by the monitoring node according to the ranking of $k$ nodes’ reputation value. In the whole process of consensus, CN is used to vote for the detailed transaction information received, generates voting results, and feeds back to the primary node. Simultaneously, the number of CN nodes needs to satisfy the following formula:

$$|S_{CN}| \geq 3f + 1$$  \hspace{1cm} (4)

where $f$ represents the maximum number of sets of malicious nodes in the CN that can tolerate malicious nodes. In practice, the number of consensus nodes $|S_{CN}|$ takes $3f + 1$.

**Primary node.** The primary node refers to a single node that is randomly selected by the monitoring node in the consensus node according to the node’s credit value and among the top half of the consensus nodes. It is used to distribute requests, write accurate transaction information statistics to new blocks, collect verification vote results for preparation blocks, and store results for a new block.

**Secondary node.** The secondary node (SN) is a storage node other than the consensus node and the monitoring node. It does not participate in the consensus process of blockchain, but only has the right to store formal blocks.

### DPBFT algorithm consensus process

In the $r$ round, the consensus process of the DPBFT algorithm is shown in Figure 4.

There are three main stages: “Pre-Prepare,” “Feedback,” and “Confirm.” The specific contents are as follows:

Step 1: the monitoring node $C$ verifies whether the node $R_i$ is the first joining

$$\begin{cases} B^0_{\text{hash}} \neq \text{hash}(0), & \text{the node join for the first time} \\ B^0_{\text{hash}} = \text{hash}(0), & \text{the restarted node} \end{cases}$$  \hspace{1cm} (5)

where $B^0_{\text{hash}}$ is the first block hash value of the node $R_i$, and $\text{hash}(0)$ is the hash256 value of character “0.” At the same time, it is judged whether the block height $B^r_{\text{high}}$ of the node $R_i$ needs synchronization

$$\begin{cases} B^r_{\text{high}} \neq B^r_{\text{maxHeight}}, & \text{the node needs to synchronize to the node of the highest block $B^r_{\text{maxHeight}}$ of the r round} \\ B^r_{\text{hash}} = \text{hash}(0), & \text{the node do not need synchronization} \end{cases}$$  \hspace{1cm} (6)

Step 2: the monitoring node $C$ calculates the reputation value of each node according to the contribution of the node $R_i$ to the system, divides the consensus node $S_{CN}$ and the secondary node $S_{SN}$, and selects the primary node $S_{CNp}$. Then, the monitoring node $C$ broadcasts the primary node $S_{CNp}$ to the whole network.

Step 3: the primary node $S_{CNp}$ obtains the transaction data $T_x$ from the sorted cache database and generates a preparation block $<\text{Pre-Prepare}, pr, h, T_x, t, p >$ into the “Pre-Prepare” phase, where $pr$ is the pre-hash of the new block, $h$ is the latest block height, $t$ is the timestamp for creating the preparation block, and $B_{pr}$ is the preparation block created by the primary node in the $r$th round.

Step 4: the primary node $S_{CNp}$ broadcasts the preparation block $B_{pr}$ to the consensus node $S_{CN}$, and the consensus node accepts the preparation block.

Step 5: the primary node $S_{CNp}$ and the consensus node $S_{CN}$ start to verify the validity of the preparation block $B_{pr}$

$$\begin{cases} \text{pre-hash}_{B_{pr}} = \text{hash}_{\text{MaxHeight}} \\ B^r_{\text{Height}} = B^{r-1}_{\text{Height}} + 1 \end{cases}$$  \hspace{1cm} (7)

where $\text{pre-hash}_{B_{pr}}$ is the pre-hash of the preparation block $B_{pr}$, $\text{hash}_{\text{MaxHeight}}$ is the hash of the highest local block of the consensus node $S_{CN}$, and $B^r_{\text{Height}}$ is the height of the previous round block.
The legal preparation block allows the consensus node $S_{CN}$ to vote, returning the result $<\text{Feedback}, pr, h, T_v, t, i, \delta_i>$ to the primary node, and entering the “Feedback” phase. The primary node $S_{CNp}$ counts the preparation block $B_{pr}$ voting result $V_{Total}$

$$V_{Total} \begin{cases} \geq \frac{k - 1}{3} + 1, \text{ vote through} \\ < \frac{k - 1}{3} + 1, \text{ voting failed} \end{cases}$$

(8)

If the number of transactions passed in the preparation block $B_{pr}$ is 0, then the primary node $S_{CNp}$ notifies the monitoring node $C$ to open the $r + 1$ round.

Step 6: the primary node $S_{CNp}$ generates the voting result into the formal block $\ll \text{Confirm, pr, h, T_v, t, hash}_{B_{pr}, p, t, \delta_p, B_r, }>$ and enters the “Confirm” phase, where $\text{hash}_{B_{pr}, p, t, \delta_p, B_r}$ is the preparation block hash generated by the $r$th primary node, and $B_r$ is the formal block. The primary node $S_{CNp}$ directly stores the formal block $B_r$ to the secondary node $S_{SN}$, broadcasts the hash of the valid transaction of the formal block to the consensus node $S_{CN}$, and then the consensus node $S_{CN}$ generates a formal block storage according to the correct transaction hash of the local preparation block $B_{pr}$ information.

Step 7: the consensus node $S_{CN}$ and the secondary node $S_{SN}$ return the stored result $\text{Store}(B_r)$ state to the primary node $S_{CNp}$, and the primary node deletes the corresponding transaction data in the cache database

$$\text{Store}(B_r) = \begin{cases} \text{true, storage completed} \\ \text{false, storage failure} \end{cases}$$

(9)

Step 8: the primary node $S_{CNp}$ sends the storage completion to the monitoring node $C$.

Step 9: the monitoring node $C$ calculates and updates the reputation value of each node, and starts the $r + 1$ round consensus.

The failure of each of the above steps will lead to the collapse of the $r$-round consensus. The unconfirmed trades in the $r$ round will be put into the $r + 1$ round to continue the consensus, and it is no longer possible to determine the list of invalid transactions directly deposited into each node $R_i$.

Figure 4. Interaction process of DPBFT consistency protocol.
It can be seen from the consensus process of the above r round that under the same n nodes, the communication complexity of DPBFT is reduced to $O(n)$. The total number of broadcast executions of the communication process is

$$f_{DPBFT} = \begin{cases} 4k + 2m - 4 & \text{if } k + m = |R| = n; \\ 3.2n - 4 & \text{if } n \geq 2 \\ 0 & \text{if } k = 0.6n, m = 0.4n \end{cases}$$

Comparing the total number of communication broadcasting executions between DPBFT and PBFT, the relationship can be obtained, as shown in Figure 5. With the increase in the number of nodes, the increase rate of the number of communication broadcasts generated by PBFT is significantly higher than that of DPBFT.

**Node credit value calculation**

The node credit value refers to the comprehensive capability of the nodes participating in the maintenance of the blockchain and plays a vital role in the evaluation of the node quality in the system. The credit-based election scheme is the core of the DPBFT algorithm. The reputation value election algorithm is shown in Table 1.

The calculation of the credit value needs to satisfy the following formula

$$f_{RV} = a + \frac{\sum_{i=1}^{2} A_i \times W_i}{\sum_{i=1}^{3} A_i} + W3 \times e^{-A_1}$$

where $f_{RV}$ represents the credit value of the node; $a$ is the initial credit value constant, which represents the initial default reputation constant of the system. The general system default is 10; $A_i$ describes the $i$th contribution of the node to the blockchain system ($A_1$ is the hardware performance memory value of the node; $A_2$ is the normal running time value of the node access system, and its dimension is minutes; and $A_3$ is the node’s accounting ability value, that is, the node storage block time, and its dimension is seconds); and $W_i$ represents the $i$th type capability weight value of the node, and its size is set according to the importance of the $i$th type capability of the node in the consensus process. The general system defaults $W_1 = 0.1$, $W_2 = 0.75$, and $W_3 = 0.15$.

To prevent the waiting time of the primary node from being too long, when the primary node counts the voting result of the preparation block and statistics the result of storing the formal block, the system sets a waiting time $t^0$. The reputation value $f'_{RV}$ of the node is

$$f'_{RV} = \begin{cases} f_{RV} - 0.2, & |t^2 - t^1| > t^0 \\ f_{RV}, & |t^2 - t^1| = t^0 \\ f_{RV} + 0.2, & |t^2 - t^1| < t^0 \end{cases}$$

Table 1. Reputation value calculation algorithm.

| Algorithm: calculate reputation value |
|--------------------------------------|
| Input: $A_i, W_i, a, t^0, t^1, t^2$ |
| Output: $f_{RV}$ |
| 1. Set initial reputation value constant $a$. |
| 2. Set initial primary node waiting for collection result time $t^0$. |
| 3. Get storage result time $t^1$ of each node to the formal block from the database. |
| 4. Get the voting result time $t^2$ of each node to the formal block from the database. |
| 5. Get capacity value of node contribution $W_i$ from the database. |
| 6. If $|t^2 - t^1| > t^0$ then |
| 7. Return $f_{RV} = f_{RV} - 0.2$. |
| 8. Else if $|t^2 - t^1| < t^0$ then |
| 9. Return $f_{RV} = f_{RV} + 0.2$. |
| 10. Else |
| 11. Return $f_{RV} = f_{RV}$. |
| 12. End if |

To show the relationship, the number between PBFT and DPBFT.
Node status change

The node status change refers to the state change of the primary node, CN, and SN, and is mainly related to the credit value of each node. The node status change can be represented in Figure 6, and the specific changes are as follows:

1. Modification of the primary node

\[
\{CN \rightarrow SN\} : CNp \in \{ f \parallel CN_p \xrightarrow{\text{info}} C = \text{error} \} \tag{13}
\]

where \( CN_p \xrightarrow{\text{info}} C \) is the information sent by the primary node to the monitoring node. At this point, the monitoring node will open a new round. The last round of unfinished transactions is saved in the cache database waiting for the new primary node to process.

2. Modification of the consensus node

\[
\{CN \rightarrow CNp\} \times \{CN \rightarrow CNp\} \xrightarrow{\text{random}} \left\{\frac{1}{2} \text{Desc} \left( f_{RV}^{\rightarrow k}(SCN) \right) \right\} \tag{14}
\]

where \( CN \xrightarrow{\text{info}} C \) is the information sent by the consensus node to the monitoring node. \( \text{Desc}(f_{RV}^{\rightarrow k}(SCN)) \) is the consensus node \( SCN_j \) to \( SCN_k \) ranked in descending order.

3. Modification of the secondary node

\[
\{SN \rightarrow CN\} : f_{RV}(t, SCN) < f_{RV}(t, SN) \tag{15}
\]

where \( f_{RV}(t, SCN) \) and \( f_{RV}(t, SN) \) represent the reputation values of the consensus node \( SCN \) and the secondary node \( SN \) at a time \( t \), respectively.

4. Modification of the newly registered node

\[
\{RN \rightarrow SN\} : \begin{cases} f_{RV}(SRN) \leq f_{RV}(SN) \\ f_{RV}(SRN) = b, b > a \end{cases} \tag{16}
\]

wherein, when a node is registered as a node in the consortium blockchain, the system assigns a reputation value \( b (b > a) \) to it as an SN by default.

Simulation test

Test platform design

The DPBFT algorithm proposed in this article is written in the Java language. The high-performance gRPC remote procedure calls (gRPC) communication protocol and the Spring Framework ensure the efficiency of service publishing and system maintainability and build a blockchain prototype system experimental platform. The experimental platform is divided into a transaction module, a consensus module, and a storage module. The transaction module sends the transaction to the consensus module; the consensus module uses the DPBFT algorithm proposed in this article, which is responsible for the consensus confirmation transaction data and the generated block sent by the transaction module; the storage module is responsible for storing the consensus made blocks in each node. The local database provides external services such as transaction data and block queries. The bottom layer of the system still uses the blockchain technology architecture system, elliptic curve cryptography (ECC) asymmetric encryption algorithm, and so on.

Experimental environment

The experimental environment of the algorithm is based on a distributed local area network with seven nodes, including one monitoring node, four consensus nodes, and two secondary nodes, which are deployed on the physical machine of the Windows 7 system, respectively. The monitoring node is Intel® Core™ i5-5200M CPU @ 2.20 GHz and 8 GB memory. The remaining nodes are Intel® Core™ i5-6500M CPU @ 3.20 GHz and 8 GB memory. \( a, W_1, W_2, W_3 \) use the system default coefficient.

Since the latest version of the Hyperledger Fabric based on the open-source project does not have a PBFT consensus algorithm, this article selects Fabric 0.6 to perform experiments in PBFT mode. The experimental environment is deployed in the same virtual machine of seven physical machines, where the order and the six peer nodes are 4 GB of memory and 100 GB of disks.

Experiment analysis

After a period of simulated transactions on the test platform of the algorithm, according to the reputation...
value formula, the monitoring node counts the reputation value of six nodes, as shown in Figure 7.

Each node accesses the system and runs normally, and the reputation value changes continuously every round. The choice of CN and SN for the monitoring node varies with the change in reputation value. The top 60% of the nodes at the same time are CN, and the last 40% of the nodes are SN. Therefore, in the CN, the election of the primary node is not constant. Peaks and valleys represent the reputation values of the two nodes when voting and storing the fastest and slowest, respectively.

In the real-time transaction of energy Internet, throughput (transactions per second (TPS)), latency, and CPU consumption play a decisive role. Therefore, to verify the optimization of the DPBFT algorithm, the corresponding performance of PBFT algorithm is compared.

The relationship between the block size and TPS. In the blockchain system, the speed of consensus affects the size of its throughput (TPS), that is, the faster the consensus speed, the higher the TPS. TPS is the number of transactions that the system can process per second, and it is an essential indicator for measuring the processing power of the blockchain system. The defined TPS formula is as follows

\[
TPS_{\Delta t} = \frac{\text{SumTrans}_{\Delta t}}{\Delta t}
\]

where \(\Delta t\) is the time when the transaction is sent to the block confirmation, and \(\text{SumTrans}_{\Delta t}\) is the total number of transactions included in the block during that time.

In this article, the actual TPS value is tested by setting the upper limit of the transaction processed by each block. The transaction module continuously sends transaction data to the Redis cache database. Take 10 blocks with different numbers of transactions, one block of \(\text{SumTrans}_{\Delta t}\) contains 1, 100, 300, 500, 1000, 1500, 2000, 2500, and 3000 transactions, respectively. In addition, 20 TPS test data were obtained for each block. Finally, according to the average TPS of 10 blocks with a different number of transactions, the TPS comparison chart of DPBFT algorithm and PBFT algorithm is drawn with MATLAB, as shown in Figure 8.

It can be seen that the TPS value of the DPBFT algorithm proposed in this article is about 4920, and the TPS of the Fabric is about 1600. The comparison shows that the TPS of the DPBFT algorithm is significantly higher than the traditional PBFT algorithm, which is enough to enhance the consensus speed of the nodes.

The test found that TPS is related to the total number of transactions included in the block: when a block contains less than a certain amount of transactions, the total number of transactions included in the block increases, and the number of transactions that can be processed per unit time increases, and the TPS is larger; when the total number of transactions included in the block exceeds a certain amount of transactions, the number of transactions in the block exceeds the processing capacity of one node, causing the thread to block and the TPS to decrease.

The relationship between the block size and latency. The TPS of the blockchain is closely related to the latency. The latency of the blockchain reflects the communication performance of the whole system and the running time of the consensus algorithm. The latency tested in this article is the time interval from the first transaction to the last transaction confirmation. The calculation formula is
\[ \text{Latency}_{time} = \frac{\text{Time}^{\text{confirm}}}{\text{Time}^{\text{generate}}} \]  
(18)

where \( \text{Latency}_{time} \) is the latency time, \( \text{Time}^{\text{confirm}} \) is the previous transaction confirmation time in the block, and \( \text{Time}^{\text{generate}} \) is the time for the first transaction.

Similarly, the above 10 blocks with different numbers of transactions are tested. Each block is tested 20 times. Finally, take the average latency of each block, and draw a comparison of the latency between the DPBFT algorithm and the PBFT algorithm. As can be seen from Figure 8, the latency of the DPBFT algorithm proposed in this article is significantly lower than that of the Fabric, which is enough to reflect that the consensus speed of the node is faster than that of the Fabric. It can also be concluded that blocks containing different numbers of transactions can affect TPS and latency. The more transactions a block contains, the larger the block, and the longer the network propagation and verification between nodes. Moreover, due to the limited network bandwidth, the longer the broadcast confirmation time, the higher the latency.

**Relationship between the number of nodes and TPS and latency.** The number of different nodes in the blockchain also affects system performance. Therefore, this section will test the system performance under different node counts of the DPBFT algorithm. As shown in Figures 9 and 10, the latency and TPS of different nodes in a block containing 300 and 500 transactions are tested on a computer under the same conditions, respectively. Experiments show that in the same size block, the latency increases as the number of nodes increases, and TPS decreases as the number of nodes increases. This is because the number of nodes...
increases, the number of communications between consensus nodes increases, and the time for data to agree on a protocol is extended, and the transaction speed of the same amount of transactions is reduced.

**Comparison of CPU usage.** In the blockchain system, the CPU usage can adequately reflect the system’s performance. Therefore, under the six nodes, this section tests the CPU usage of the DPBFT algorithm and the PBFT algorithm, as shown in Figure 11.

From the results, it can be concluded that the average CPU usage of DPBFT is about 13.57%, while the average CPU usage of PBFT is approximately 21.59%. Therefore, the DPBFT algorithm reduces the use of computing power to a certain extent, thereby effectively reducing transaction time.

**Conclusion**

Blockchain technology is a research hotspot in recent years, and many industries are using it for innovation. Because different industries have different characteristics, a consensus algorithm is also difficult to meet all industries. In this article, the traditional PBFT consensus algorithm is studied and improved according to the demand for low-latency transaction applications in the energy field. Therefore, a DPBFT algorithm is proposed, which is a method of introducing a reputation value in the traditional PBFT algorithm to divide nodes with different permissions, so as to avoid all nodes from participating in consensus operations. Furthermore, in order to reduce the communication complexity of the original PBFT, the consensus process confirms the legitimacy of the block through the primary node. Thereby, this article uses the DPBFT algorithm to solve the problems of a long consensus period, low efficiency, and high CPU usage in traditional PBFT consensus algorithm. Finally, the feasibility and availability of the improved consensus mechanism for real-time transactions in the energy field were verified by building an experimental platform.

Meanwhile, DPBFT proposed in this article also has some shortcomings, such as the singleness of the monitoring node, which makes it challenging to ensure that the monitoring node has not been down. Therefore, in future research, more research is needed on whether the monitoring node can solve this problem through the cluster.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Scientific Research Foundation of Kunming University of Science and Technology (KKSYS201703016), Yunnan Young and Middle-aged Academic and Technical Leaders Reserve Talents (Shen Tao, 2018), Yunnan Provincial Ten Thousands Plan Young Top Talents (KKRD201903071, Yunren Social Development No. 2018 73), and State Grid ICT Industry Group Two-level Collaborative Research and Development Project in 2019.

**ORCID iD**

Wenjun Cai https://orcid.org/0000-0001-9551-2356

**References**

1. Haring TW, Mathieu JL and Andersson G. Comparing centralized and decentralized contract design enabling direct load control for reserves. *IEEE T Power Syst* 2016; 31(3): 2044–2054.
2. Dunnan L, Tianqi T, Jianhua Y, et al. Energy Internet based micro balance dispatching and trading design. *Autom Electr Power Syst* 2017; 41(10): 1–8, 36.
3. Mengelkamp E, Notheisen B, Beer C, et al. A blockchain-based smart grid: towards sustainable local energy markets. *Comput Sci Res Develop* 2018; 33(1–2): 207–214.
4. Lin CC, Deng DJ, Kuo CC, et al. Optimal charging control of energy storage and electric vehicle of an individual in the Internet of energy with energy trading. *IEEE T Ind Inform* 2018; 14(6): 2570–2578.
5. Dong Z, Luo F and Liang G. Blockchain: a secure, decentralized, trusted cyber infrastructure solution for future energy systems. *J Modern Power Syst Clean Energy* 2018; 6(5): 118–127.
6. Aitzhan NZ and Svetinovic D. Security and privacy in decentralized energy trading through multi-signatures,
blockchain and anonymous messaging streams. *IEEE Trans Depend Secure Comput* 2018; 15(5): 840–852.

7. Ming Z, Jun C, Yuqing W, et al. Primarily research for multi module cooperative autonomous mode of energy Internet under blockchain framework. *Proc CSEE* 2017; 37(13): 3672–3681.

8. Sharma PK, Kumar N and Park JH. Blockchain-based distributed framework for automotive industry in a smart city. *IEEE Trans Ind Inform* 2018; 15: 4197–4205.

9. Ding D and Xuantong Z. Blockchain technology and its prospect of industrial application. *J Inform Secur Res* 2017; 3: 200–210.

10. Li Z, Kang J, Yu R, et al. Consortium blockchain for secure energy trading in industrial Internet of things. *IEEE Trans Ind Inform* 2017; 14: 3690–3700.

11. Shao QF, Jin CQ, Zhang Z, et al. Blockchain: architecture and research progress. *Chin J Comput* 2018; 41(5): 969–988.

12. Ding W, Wang GC, Xu AD, et al. Research on key technologies and information security issues of energy blockchain. *Proc CSEE* 2018; 38: 1026–1034.

13. Gai K, Wu Y, Zhu L, et al. Privacy-preserving energy trading using consortium blockchain in smart grid. *IEEE Trans Ind Inform* 2019; 15: 3548–3558.

14. Yang D, Zhao X, Xu Z, et al. Developing status and prospect analysis of blockchain in energy Internet. *Proc CSEE* 2017; 37(13): 3664–3671.

15. Christidis K and Devetsikiotis M. Blockchains and smart contracts for the Internet of things. *IEEE Access* 2016; 4: 2292–2303.

16. Fan T, He Q, Nie E and Chen S. A study of pricing and trading model of blockchain & big data-based energy-Internet electricity. In: *Proceedings of the 3rd international conference on environmental science and material application (ESMA 2018)*, Chongqing, China, 25–26 November 2018, pp.1–12. New York: IEEE.

17. Claudia P, Tudor C, Marcel A, et al. Blockchain based decentralized management of demand response programs in smart energy grids. *Sensors* 2018; 18(2): 162.

18. Hinterstocker M, Dufter C, von Roon S, et al. Potential impact of blockchain solutions on energy markets. In: *Proceedings of the 15th International Conference on the European energy market (EEM)*, Lodz, 27–29 June 2018, pp.1–5. New York: IEEE.

19. Wang J, Wang Q and Zhou N. A decentralized electricity transaction mode of microgrid based on blockchain and continuous double auction. In: *Proceedings of the 2018 IEEE power & energy society general meeting (PESGM)*, Portland, OR, 5–10 August 2018, pp. 1–5. New York: IEEE.

20. Wang J, Zhou N, Wang Q, et al. Electricity direct transaction mode and strategy in microgrid based on blockchain and continuous double auction mechanism. *Proc CSEE* 2018; 38(17): 5072–5085.

21. He Q, Xu Y, Yan Y, et al. A consensus and incentive program for charging piles based on consortium blockchain. *CSEE J Power Energy Syst* 2018; 4(4): 452–458.

22. Pipattanasomporn M, Kuzlu M and Rahman S. A blockchain-based platform for exchange of solar energy: laboratory-scale implementation. In: *Proceedings of the 2018 international conference and utility exhibition on green energy for sustainable development (ICUE)*, Phuket, Thailand, 24–26 October 2018, pp.1–9. New York: IEEE.

23. Di Silvestre ML, Gallo P, Ippolito MG, et al. A technical approach to the energy blockchain in microgrids. *IEEE Trans Ind Inform* 2018; 14(11): 4792–4803.

24. Thakur S, Hayes BP and Breslin JG. Distributed double auction for peer to peer energy trade using blockchains. In: *Proceedings of the 2018 5th international symposium on environment-friendly energies and applications (EFEA)*, Rome, 24–26 September 2018, pp. 1–8. New York: IEEE.

25. Zheng Z, Xie S, Dai H, et al. An overview of blockchain technology: architecture, consensus, and future trends. In: *Proceedings of the 2017 IEEE international congress on big data (Bigdata Congress)*, Honolulu, HI, 25–30 June 2017, pp. 557–564. New York: IEEE.

26. Zheng Z, Xie S, Dai HN, et al. Blockchain challenges and opportunities: a survey. *Int J Web Grid Serv* 2017; 14: 5.

27. Mingxiao D, Xiaofeng M, Zhe Z, et al. A review on consensus algorithm of blockchain. In: *Proceedings of the 2017 IEEE international conference on systems, man, and cybernetics (SMC)*, Banff, AB, 5–8 October 2017, pp. 2567–2572. New York: IEEE.

28. Reiter M. A secure group membership protocol. *IEEE Trans Softw Eng* 1996; 22(1): 31–42.

29. Castro M and Liskov B. Practical Byzantine fault tolerance. In: *Proceedings of the third symposium on Operating systems design and implementation*, New Orleans, LA, 22–25 February 1999. New York: ACM.

30. Zhang Y, Yu R, Nekovee M, et al. Cognitive machine-to-machine communications: visions and potentials for the smart grid. *IEEE Network* 2012; 26(3): 6–13.