An Efficient Blockchain PBFT Consensus Protocol in Energy Constrained IoT Applications

Xiaoqiong Xu
The School of Information and Communication Engineering, University of Electronic Science and Technology of China,
Chengdu, China
xiaoqiongxu810@gmail.com

Gang Sun
The School of Information and Communication Engineering, University of Electronic Science and Technology of China,
Chengdu, China
gangsun@uestc.edu.cn

Hongfang Yu
The School of Information and Communication Engineering, University of Electronic Science and Technology of China,
Chengdu, China
yuhf@uestc.edu.cn

2021 International Conference on UK-China Emerging Technologies (UCET) | 978-1-6654-9575-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/UCET54125.2021.9674967

***Abstract*—While the Internet of Things (IoT) can support various smart applications, data privacy and security still have been the problems due to a great diversity of IoT devices. Blockchain technology, with distributed, traceable and non-tamperable characteristics, offers opportunities to solve these problems. However, due to the limited energy and computing capability of IoT devices, it is hard to conduct expensive consensus in blockchain systems, especially in PoW based blockchains. In this paper, we propose a high energy efficiency PBFT consensus protocol designed for energy-constrained IoT-blockchain applications. In our proposed PBFT protocol, we design the energy effective consensus node selection mechanism and use the VRF to ensure the security of the leader. Besides, we investigate the node authority by extending the degree of centrality to choose relay nodes in the case of multi-hop neighbor nodes. The simulation results validate that our proposed consensus protocol can cut down the energy consumption as well as can attain high performance.**

***Keywords*—Blockchain, PBFT Consensus, Energy Effective, Internet of Things (IoT)**

I. Introduction

Internet of Things (IoT) is rapidly gaining popularity from both industry and academia [1]. The IoT network aims to connect massive devices and sensors which produce different kinds of data to support new day-to-day services and business methods. However, the centralized IoT topology and the resource constraints of heterogeneous IoT devices bring new serious data privacy and security problems in large-scale IoT adoptions. Nowadays, blockchain has been regarded as one of the most promising approaches to address these security issues owing to its capabilities such as decentralization, transparency, anonymity and data encryption. Blockchain [2], widely used in cryptocurrencies, is fundamentally a distributed immutable ledger that stores and shares transactions across a peer-to-peer (P2P) network. In which, individuals can interact with others in a verifiable manner without the involvement of any trusted third party. Some existing research are already using blockchain technology in many IoT sectors, including Smart Cities, Smart Homes, Healthcare, Smart grid, etc [3], [4].

Although good prospects can be expected, it should also be noted that the combination of blockchain and IoT faces some new challenges. We know that the majority of blockchain systems use Proof of Work (PoW) as their underlying consensus mechanism, in which miners compete to produce new blocks with a complex mathematical computation [5]. However, most IoT devices are mobile devices or sensors which have limited computing capability, it is hard for them to conduct expensive mining work. Meanwhile, PoW consensus protocol typically suffers from very poor performance with a latency of tens of minutes, thus inefficient for delay-sensitive IoT applications. In contrast, Practical Byzantine Fault Tolerance (PBFT) [6] consensus protocol reaches high performance consensus through the message transmission mechanism and quorum theory in an asynchronous model. It works on the assumption that less than one-third of the peers are faulty. While, due to the low computational power and complexity, PBFT is deemed more suitable for consensus in IoT-blockchain systems.

Unfortunately, adopting PBFT for consensus in IoT-blockchain systems is not straightforward and will face the following critical challenges. Firstly, the PBFT protocol relies on all-to-all internode communications which result in the exponentially increasing message ($O(n^2)$ message complexity among n nodes). Thus, it scales poorly as the number of nodes in the network increases, this is contrary to IoT networks which are expected to involve numerous devices. Secondly, PBFT puts a high burden on the leader node who needs to propose messages of large size to all nodes, making the leader a bottleneck. The failure of the leader node would trigger the view change which is a slow and costly process. In this case, the performance will be exponentially dropped, thus cannot well satisfy the requirement of IoT applications.

Many scholars proposed some improvement on PBFT protocol, these protocols investigated how to make the leader rotation for alleviating bottlenecks in BFT protocols. However, the improved PBFT protocols are also vulnerable to DoS attacks, are low consensus performance, and thus are not well-suited for the energy-constrained IoT applications. In this

This work was supported by the PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications PCL2018KP001. (Corresponding author: Hongfang Yu.)

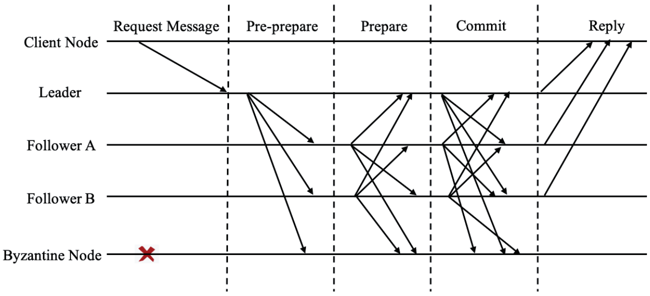

Fig. 1. The workflow of PBFT involves three phases: Per-prepare, Prepare, and Commit.

paper, we introduced a security-guaranteed PBFT protocol called S-PBFT. We use the available energy of nodes as a threshold to construct the consensus node group. Moreover, at the beginning of each round of consensus, the verifiable random function (VRF) is applied to enable the security of the leader. Then, degree centrality is used to elect the internodes to reach low communication overhead. Only the nodes with the higher degree are used to forward messages instead of all-to-all in traditional PBFT.

The rest of this paper is organized as follows. Section II presents the background of traditional PBFT consensus protocol and related works. Section III details the design of our S-PBFT. Section IV evaluates the performance of our protocol by comparing it with traditional PBFT. Finally, Section V concludes our work.

II. Background

To describe our proposed protocol, this section first introduces the workflow of traditional PBFT. Then the existing improved PBFT protocols in blockchain systems are given.

A. PBFT Consensus

PBFT protocol provided an effective solution to Byzantine Problem, which was intractable and widely concerned in distributed computing systems. It can work correctly and effectively when the number of malicious (or dishonest) nodes is less than a third of the total network nodes. In PBFT, one of the nodes is selected as the primary node (also called leader), while others are follower nodes. All nodes in the system interact with others to reach an agreement on the system's state.

The workflow of PBFT protocol can be divided into three phases, that is the pre-prepare phase, prepare phase, and commit phase, as shown in Figure 1.

1) In the pre-prepare phase, after received a request message from the client node, the leader constructs a pre-prepared message which includes the encoded request message. Then, the leader broadcasts it to all follower nodes.

2) If a follower node receives and accepts the pre-prepare message after verification, it enters the prepare phase. The verification is used to ensure that this pre-prepare message is from the leader. If the verification is successful, the follower node will broadcast the prepared message to the other nodes. Meanwhile, it also receives the prepared message from others, and checks them validity. When a follower node gets $2f$ valid prepared messages from different follower nodes, the prepare phase is finished, where f is the number of Byzantine nodes in the system.

3) In the commit phase, each node broadcasts a commit message to others for validation. Once the number of received commit messages (including itself) is equal or exceeds $2f+1$, the node will send a reply message to the client. The consensus is achieved after the client collected the same $f+1$ reply messages.

B. Improved PBFT Protocols

Many scholars have made improvements on traditional PBFT protocol from various aspects, which can be classified into two categories: consensus nodes election and communication optimization.

(1) Consensus nodes election: Gao *et al.* [7] exploited the node trust which was obtained by the transactions between nodes to propose a novel T-PBFT protocol. In which, the high trustworthy of nodes were selected as consensus nodes. And, to reduce the probability of view change, it replaced a single leader in PBFT with a leader group. Similarly, SG-PBFT [8] used scores to select suitable consensus nodes. After each round of consensus, the scores of all nodes were updated by their judgment result in this round. Nodes with high scores would be selected as consensus nodes in the next round of consensus. In addition, Lao *et al.* [9] proposed a scalable PBFT consensus algorithm that selected the fixed nodes as the consensus participants to ensure security. This is because, compared with mobile nodes, fixed nodes have the greater computing power and thus have a lower probability of becoming malicious nodes. Besides, Li *et al.* [10] proposed Extensible-PBFT (EPBFT), which implemented the selection of consensus nodes with verifiable random function (VRF) to ensure fairness and security.

(2) Communication optimization: Onireti *et al.* [11] proposed to reduce the number of duplicate nodes in the PBFT by defining feasible areas for wireless networks, thereby reducing system overhead and improving system efficiency. Feng *et al.* [12] proposed a scalable dynamic multi-agent PBFT consensus (SDMA-PBFT) to reduce the communication cost. The main idea was to use hierarchical and agent technology to divide the system into multi-layer. Each layer included various areas in which the agent node was served as the leader when conducts PBFT consensus. Wang *et al.* [13] proposed to improve the communication topology of the blockchain network using a star network structure. The optimization was focused on the commit phase of PBFT, in which the replica node only communicated with the leader, and the leader node determined the final decision value based on all collected information.

The studies mentioned above did not consider the energy problem when assigning a node as leader or consensus node. Thus, they are not feasible in energy-constrained IoT environment.

III. S-PBFT Consensus Protocol Details

In this section, we present details on our improved S-PBFT consensus protocol, which includes energy-efficient consensus nodes selection, secure leader selection, and optimized message transmission.

A. Energy-efficient Consensus Nodes Selection

In actual IoT applications, most nodes are typically energy-constrained. If the available energy is very low (can't support consensus consumption), nodes will be byzantine Failure or fail-stop fault. It's much harder to achieve consensus in a distributed system when there are many faulty nodes. Thus, we propose an energy efficiency strategy to form a consensus node group. This consensus node group is updated periodically.

In our proposed strategy, the nodes with high available energy (more than the energy threshold E^{th}) are selected as a consensus node to participate in the consensus, that is to produce blocks, validate and transmit consensus messages according to the PBFT protocol. And the node with low available energy ($\leq E^{th}$) are selected as candidate nodes that only update their local ledger states when reaching a consensus. The candidate nodes don't participate in the consensus process until they are rechanged.

Assume that the selection cycle is T, the block interval is τ. N_{packet} is the average number of packets of the consensus message. While there are N nodes in the IoT system and the allowed Byzantine nodes can not exceed f. The number of consensus nodes CN must satisfy $CN \geq 3f+1$ to ensure that the system can reach an agreement in any case. The energy threshold E^{th} of the consensus nodes selection depends on two factors: (1) the energy required to transmit the consensus messages (E_{tx}); and (2) the energy required for the nodes to run its basic functions (such as scanning the radio environment)(E_{scan}). Thus, the energy threshold of the IoT nodes can be calculated as follows.

$$E^{th} = E_{tx} + E_{scan} \tag{1}$$

and,

$$E_{tx} = N_{packet} \times E_{tx\_per\_packet} \times \lceil \frac{T}{\tau} \rceil \times 2(CN-1) \tag{2}$$

$$E_{scan} = T \times E_{tx\_scan} \tag{3}$$

where $E_{tx\_per\_packet}$ is the energy consumed by transimtting one packet, and E_{tx_scan} is the energy consumed for scanning to the radio environment per second.

Certainly, we know that the continuous charging IoT devices (such as fixed base stations) will always participate in consensus. To a certain extent, this type of IoT device also has better safety performance, thus is reasonable that these nodes would behave honestly at a high probability.

B. Secure Leader Selection

Leader selection is the first step of each round of consensus, which is aimed to designate a leader to propose a new block to follower nodes. The key challenge is to keep the selection result unpredictable, that is an adversary does not know the leader until the selected leader starts to propose the new block. Verifiable Random Function (VRF) is a cryptographic function that processes inputs to verifiable pseudorandom outputs. By using the proof and the public key as inputs, the outputs can be verifiable without knowing the private key and even without the possibility to find out it. Thus, it is suitable to be used to choose leaders to propose a block and defend the DoS attack. Below, we present details on the leader selection performed by our S-PBFT which is based on the VRF.

To allow a user to prove that it was chosen as leader, each consensus node i has a public/private key pair (pk_i, sk_i). Consensus node i puts a seed x and its private key sk_i as the inputs of the *VRF* function. This seed x is publicly known by other consensus nodes for each round, but cannot be controlled by the adversary. The seed in the round t in our S-PBFT is calculated as follows,

$$x_t = (t + \text{block-number})\%\ CN \tag{4}$$

where, the block-number is the number of blocks in the local ledger.

Then, VRF returns two values: a hash ξ and a proof ρ. Node i uses the hash ξ as the input of *VRF-Proof* function to determine whether it is the leader. The result of the *VRF-Proof* function is:

$$\text{result} = \begin{cases} \text{yes}, & (H_{256}(\xi)/(2^{hashlen}) \leq \varepsilon \\ \text{no}, & (H_{256}(\xi)/(2^{hashlen}) > \varepsilon \end{cases} \tag{5}$$

where $\varepsilon(\varepsilon \in [0,\ 1])$ is the selection threshold, H_{256} is the sha-256 hash function, and the $hashlen$ indicates the length of the hash result.

If node i was selected as leader, it packages transactions into a new block and proposes per-prepare messages (with its hash ξ and proof ρ) to other consensus nodes. Node j received these messages, it uses the *VRF-Verify* function to check if that node i was selected as leader. This verification procedure includes two parts: the legality of ξ and ρ and the corresponding hash result is lower than ε. That is,

$$\text{Verify}(x, \xi, \rho, pk_i)\ \&\ (H_{256}(\xi)/(2^{hashlen}) \leq \varepsilon = 1 \tag{6}$$

If the verification passes, node j will produce a prepare message and broadcast it to other nodes. Otherwise, it will drop this message.

C. Optimized Message Transmission

The leader-based PBFT consensus protocols suffer from the high communication burden of broadcasts, resulting in larger waste of network resources. To reduce the communication complexity, in S-PBFT, we use the well-known gossip protocol over an application layer overlay network to disseminate consensus messages. Moreover, relay nodes at each hop forward messages only to its k connected neighbors to cut down the number of message copies, hence can effectively reduce network overhead. However, the message propagation speed largely depends on these k connected neighbors, randomly neighbors selection may cause high delay. To avoid this problem, we estimate the node's centrality in the network and make neighbors selection. Nodes with a higher degree of centrality keep contact with more numbers of other nodes, Such nodes can be the central nodes that occupy a structural position to act as a conduit for information exchange.

In our S-PBFT, we utilize the "closeness" centrality to evaluate the importance of nodes. "Closeness" centrality measures the reciprocal of the mean geodesic distance and the "closeness" centrality for a given node i is,

$$D_i = \frac{N-1}{\sum_{i=1}^{N} d(i,j)} \tag{7}$$

where $d(i,j) = 1$ is the shortest path between node i and node j. And N is the number of nodes in the network and $i \neq j$.

"Closeness" centrality can indicate how long it will take messages to spread from a given node to other nodes in the network. Thus, we can speed up the message propagation to select neighbors with higher "closeness" centrality.

IV. Simulation Results

To validate the performance of our S-PBFT, we conduct a series of simulation experiments to make comparisons of traditional PBFT and S-PBFT.

A. Experiment Setup

These experiments use two kinds of nodes: fixed nodes which with continuous power supply and mobile nodes with initialization energy. The mobile nodes are evenly distributed in the communication radius of the fixed nodes and are recharged with a probability of 0.05 per hour. The consensus node group consists of $10 \sim 100$ nodes, and the minimal and maximal number of misbehavior nodes is set as 1 and 5 separately to meet the requirement of $CN \geq 3f + 1$). The block interval $\tau = 10s$, the average number of packets in the consensus message $N_{packet} = 100$.

B. Evaluation Metrics

We extensively evaluate and compare our proposed S-PBFT with traditional PBFT protocol according to the following metrics.

Scalability: we measure the throughput and latency with different numbers of nodes to evaluate system scalability. The throughput refers to the number of transactions written into the ledger by the system per second. The latency is from the time when a transaction request is sent to the leader to the time when this request is written to the ledger after consensus.

Communication overhead: we use the average energy consumption in each node for one successful consensus to estimate the communication overhead, which can be defined as follows:

$$E_{ava} = \frac{\sum_{i=1}^{N} E_{sum}^{i}}{M \cdot N} \tag{8}$$

where, E_{sum}^{i} is the total energy consumption of node i, the total number of successful consensus is M.

Security: the security is guaranteed by *VRF-Verify* in Section III-B, but it is infeasible to prove it with experimental method due to the wide variety of possible attacker strategies. Thus, we design an actual DoS attack strategy, that is: a part of misbehavior nodes (50%) send an overwhelming number of requests to the leader in the last round; Other misbehavior nodes choose another potential leader to seed malicious consensus messages. The security is evaluated with the ratio of successful attacks as the number of nodes.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Number of nodes	$[10, 20, 30, 40, 50]$
Number of fixed nodes	$[1, 2, 3, 4, 5]$
Number of misbehavior nodes	$[1, 2, 3, 4, 5]$
Request submission rate	500 tps
Block Interval τ	10 s
Average number of packets N_{packet}	100
Initialization energy of the mobile nodes	[1000 mAh, 10000mAh]
$E_{tx\_per\_packet}$	0.00003 mAh
E_{tx_scan}	0.0005 mAh/s

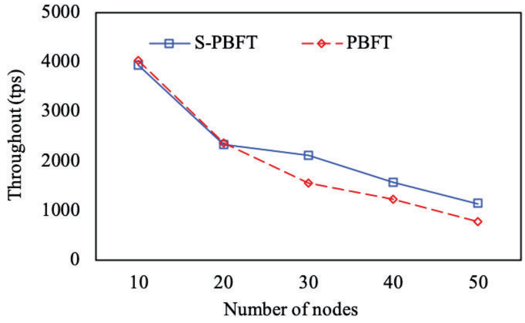

Fig. 2. The throughput versus the number of nodes.

C. The Results

We increase the number of nodes from 10 to 50 with step 10, in which the number of fixed nodes is from 1 to 5 with step 1. The initialization energy of mobile nodes is set with [1000mAh, 10000mAh]. For other parameters, refer to Table I For all experiments shown in this section, we conduct 100 runs for each parameter to eliminate possible errors and the final results are the average of these sum runs.

Throughput: Figure 2 gives the comparison of the throughput of these two consensus protocols. In this experiment, we set up the client to send 500 requests per second with different numbers of nodes. From the Figure, we know that the throughput of our S-PBFT consensus protocol is higher than that of the traditional PBFT consensus protocol. At the same time, with the increase in the number of nodes, the throughput of the two algorithms all showed a downward trend. The advantages of the S-PBFT algorithm are still very obvious.

Latency: Figure 3 shows the consensus latency of the two consensus protocols. We can find from Figure 3, the consensus

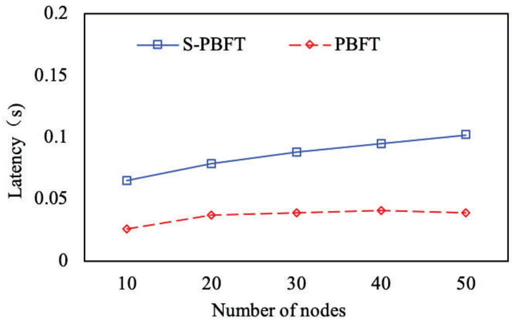

Fig. 3. The consensus latency versus the number of nodes.

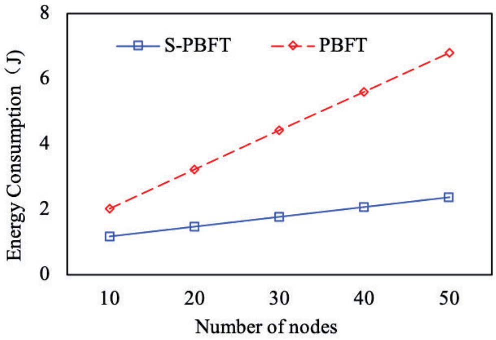

Fig. 4. The energy consumption versus the number of nodes.

latency of S-PBFT is significantly higher than that of PBFT. The main reason is that our scheme reduces the number of relay nodes to transmit messages instead of all-to-all gossip transformation in PBFT, thus slow down consensus speed.

Energy Consumption: Figure 4 presents the energy consumption for these two consensus protocols. We can see that the energy consumed in PBFT is proportional to network size due to all-to-all transmission. Messages will accordingly flood the network consuming more energy. As in S-PBFT, however, we use k neighbors to relay the message. It can lead to significantly low energy consumption, especially in larger networks.

Ratio of the Successful Attack: Figure 5 and 6 show the ratio of successful attack for these two consensus protocols. The results show that at least empirically for this particular attack, our S-PBFT is not affected. In contrast, the traditional PBFT shows a big difference as the number of nodes. Besides, the successful attack number in our S-PBFT is significantly lower than traditional PBFT (10×). This proves that our S-PBFT has better security.

V. Conclusions

The scalability and security are key challenges of a blockchain system for the energy-constrained Internet-of-Things. This paper proposes an improved S-PBFT consensus protocol by designing the energy-efficient consensus nodes selection, using VRF to obtain the leader, and adopting a degree-based communication mechanism. Numerical results validated that our proposed consensus protocol can mitigate DoS attacks and provide high-throughput under a wide number of nodes.

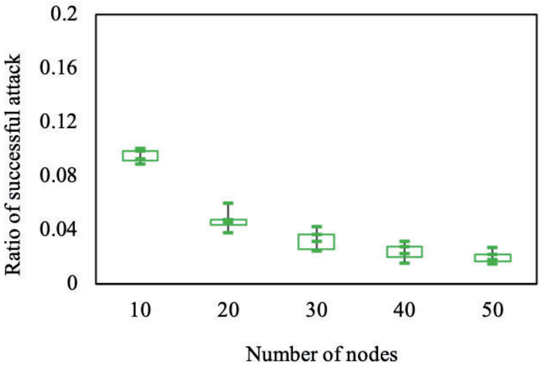

Fig. 5. The ratio of successful attack in PBFT versus the number of nodes.

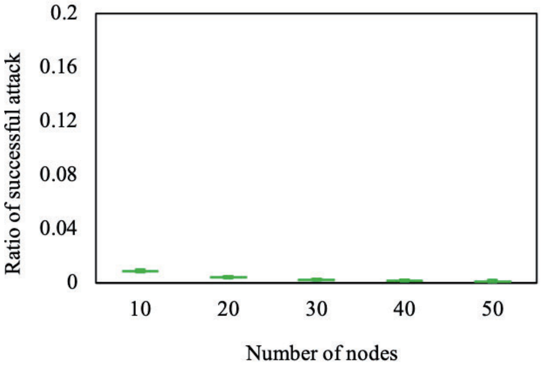

Fig. 6. The ratio of successful attack in S-PBFT versus the number of nodes.

References

[1] J. Wang, M. K. Lim, C. Wang, and M.-L. Tseng, "The evolution of the internet of things (iot) over the past 20 years," *Computers & Industrial Engineering*, vol. 155, p. 107174, 2021.

[2] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," *Information Processing & Management*, vol. 58, no. 1, p. 102436, 2021.

[3] L. Luo, J. Feng, H. Yu, and G. Sun, "Blockchain-enabled two-way auction mechanism for electricity trading in internet of electric vehicles," *IEEE Internet of Things Journal*, 2021.

[4] X. Xu, X. Wang, Z. Li, H. Yu, G. Sun, S. Maharjan, and Y. Zhang, "Mitigating conflicting transactions in hyperledger fabric permissioned blockchain for delay-sensitive iot applications," *IEEE Internet of Things Journal*, 2021.

[5] G. Sun, M. Dai, J. Sun, and H. Yu, "Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6257–6272, 2020.

[6] S. Alqahtani and M. Demirbas, "Bottlenecks in blockchain consensus protocols," *arXiv preprint arXiv:2103.04234*, 2021.

[7] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-pbft: An eigentrust-based practical byzantine fault tolerance consensus algorithm," *China Communications*, vol. 16, no. 12, pp. 111–123, 2019.

[8] G. Xu, Y. Liu, J. Xing, T. Luo, Y. Gu, S. Liu, X. Zheng, and A. V. Vasilakos, “Sg-pbft: a secure and highly efficient blockchain pbft consensus algorithm for internet of vehicles,” *arXiv preprint arXiv:2101.01306*, 2021.

[9] L. Lao, X. Dai, B. Xiao, and S. Guo, “G-pbft: a location-based and scalable consensus protocol for iot-blockchain applications,” in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2020, pp. 664–673.

[10] Y. Li, Z. Wang, J. Fan, Y. Zheng, Y. Luo, C. Deng, and J. Ding, “An extensible consensus algorithm based on pbft,” in *2019 International conference on cyber-enabled distributed computing and knowledge discovery (CyberC)*. IEEE, 2019, pp. 17–23.

[11] O. Onireti, L. Zhang, and M. A. Imran, “On the viable area of wireless practical byzantine fault tolerance (pbft) blockchain networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

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

[13] F. Wang, Y. Ji, M. Liu, Y. Li, X. Li, X. Zhang, and X. Shi, “An optimization strategy for pbft consensus mechanism based on consortium blockchain,” in *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, 2021, pp. 71–76.