LightChain: A Lightweight Blockchain System for Industrial Internet of Things

Yinqiu Liu, Kun Wang, *Senior Member, IEEE*, Yun Lin, and Wenyao Xu, *Senior Member, IEEE*

***Abstract*—While the intersection of blockchain and Industrial Internet of Things (IIoT) has received considerable research interest lately, the conflict between the high resource requirements of blockchain and the generally inadequate performance of IIoT devices has not been well tackled. On one hand, due to the introductions of mathematical concepts, including Public Key Infrastructure, Merkle Hash Tree, and Proof of Work (PoW), deploying blockchain demands huge computing power. On the other hand, full nodes should synchronize massive block data and deal with numerous transactions in peer-to-peer network, whose occupation of storage capacity and bandwidth makes IIoT devices difficult to afford. In this paper, we propose a lightweight blockchain system called *LightChain*, which is resource-efficient and suitable for power-constrained IIoT scenarios. Specifically, we present a green consensus mechanism named Synergistic Multiple Proof for stimulating the cooperation of IIoT devices, and a lightweight data structure called LightBlock to streamline broadcast content. Furthermore, we design a novel Unrelated Block Offloading Filter to avoid the unlimited growth of ledger without affecting blockchain's traceability. The extensive experiments demonstrate that *LightChain* can reduce the individual computational cost to 39.32% and speed up the block generation by up to 74.06%. In terms of storage and network usage, the reductions are 43.35% and 90.55%, respectively.**

***Index Terms*—Blockchain, consensus mechanism, distributed system, data filter, industrial Internet of Things (IIoT).**

Manuscript received October 7, 2018; revised January 27, 2019; accepted February 24, 2019. Date of publication March 11, 2019; date of current version June 12, 2019. This work was supported in part by National Natural Science Foundation of China under Grant 61872195, Grant 61572262 and in part by the U.S. National Science Foundation under Grant 1718375. Paper no. TII-18-2637. *(Corresponding authors: Kun Wang and Wenyao Xu.)*

Y. Liu is with the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: yinqiuliu@foxmail.com).

K. Wang is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles 90095 USA (e-mail: wangk@ucla.edu).

Y. Lin is with the Communication and Information College, Harbin Engineering University, Harbin 150001, China (e-mail: linyun@hrbeu.edu.cn).

W. Xu is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 USA (e-mail: wenyaoxu@buffalo.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2904049

I. Introduction

As Industrial Internet of Things (IIoT) has been globally adopted, many development bottlenecks follow. First, since IIoT system generally contains a large number of scattered devices, it is highly vulnerable when facing Distributed Denial of Service (DDoS) attacks [2]. Second, the centralized management structure cannot be self-certified. Related incidents, such as personal privacy leaking, occur occasionally. Finally, with the continuous evolution of Low Power Wide Area Network (LPWAN) [3], the number of IIoT devices will grow geometrically in the future, leading the cost of centralized services become unaffordable.

Fortunately, a growing body of recent research works have focused on deploying blockchain in IIoT to tackle the above issues. Blockchain is a decentralized, append-only ledger. Adopting consensus operation with distributed storage, blockchain technology features nonrepudiation and non-tampering [4], [5]. With its support, industrial information will be jointly maintained in a P2P network, which can effectively implement data traceability, and facilitate the value flow in nontrusted environments. Moreover, blockchain is capable of providing Distributed Decentralized Domain Name Service, contributing to mitigate current DNS vulnerabilities, including DDoS attacks, DNS spoofing, etc., [7]. For example, some well-known works, e.g., RuffChain [6], Consortium Blockchain [8], [9], Probe-IoT [10], were compromised to realize the combination between blockchain and IoT/IIoT.

However, since blockchain eliminates the reliance on certificate authorities, it applies mathematical and distributed mechanisms to ensure data security or to prevent attacks. Such settings result in extremely stringent resource requirements. More seriously, the performance of IIoT devices is generally insufficient. To optimize the resource consumption of blockchain, researchers have presented a variety of proposals. For instance, in Vegvisir [11], Karlsson *et al.* proposed a permissioned, directed acyclic graph (DAG)-structured blockchain suitable for power-constrained environments with limited network connectivity. In AlkylVM [12], Ellul *et al.* described a split-virtual machine which allows devices to interact with blockchain systems. Additionally, Liu *et al.* [13] presented computation offloading and content caching in wireless blockchain networks to handle the PoW puzzle and ever-increasing Internet traffic. While all of these systems may not thoroughly lighten the resource consumption of blockchain systems.

We observe that deploying blockchain in IIoT faces several key challenges:

1) Since proposing blocks can bring benefits, rich peers will continually improve their computing power under PoW, while most of participants cannot keep pace with such update speed. As block height increases, the impact of Matthew Effect will become serious, leading to the centralization of computing power in P2P network.
2) To achieve distributed consistency, blockchain requires that individual node must preserve massive data generated by numerous network participants. All historical data will be permanently stored in local without recycling mechanism, encroaching on a large amount of storage space.
3) IIoT devices are in the confrontation with heterogeneous network resources. Therefore, some resource-constrained peers cannot support the operations related to blockchain in the case of high throughput.

In this paper, to solve the challenges discussed above, we present the concept of *lightweight blockchain*, i.e., to minimize the blockchain-to-node resource occupancy; thus, ensure the original performance of devices. Specifically, we summarize such occupancy into three significant aspects: computing power consumption; storage space usage; and network resources usage. On top of this insight, we develop a lightweight system prototype to lighten blockchain from these sorts. Our contributions are summarized as follows:

1) We present a green consensus mechanism called Synergistic Multiple Proof (SMP) to promote cooperation between IIoT devices. Different from PoW where mining merely depends on workload, we apply multiple standards and propose the framework of Collaboration Index ($\mathcal{CI}$). Via $\mathcal{CI}$, we set the dynamic difficulty for each node, effectively saving the computing power consumption of mining.
2) Considering the limitation on storage resource, we exploit a novel Unrelated Block Offloading Filter (UBOF). Through the analysis of Unspent Transaction Output (UTXO), we propose the definition of Unrelated Blocks ($\mathcal{UB}$). UBOF can detect and offload $\mathcal{UB}$s, contributing to reduce the storage resources occupied by blockchain.
3) We observe that during transaction and block verification, the information broadcast by peer nodes exists overlap, which is not conducive to improving throughput. Hence, we design a lightweight data structure named LightBlock (LB) instead of the entire block for broadcasting.
4) We also develop a prototype of LightChain on Python 3 [14] with support from Gunicorn [15]. Evaluations in a P2P network of ten miners and numerous light-weight nodes demonstrate that SMP reduces 60.86% computational cost of miners, and speeds up the block generation by up to 74.06%. Furthermore, UBOF maintains a filtration efficiency of 43.35% under different workload, and LB saves network usage by upto 90.55%.

The structure of this paper is organized as follows. We first discuss the related work and our motivation in Section II. Then, the system model and layer functions are given in Section III. We present the detailed design of LightChain layer in Section IV, as well as the lightweight data structures and mechanisms we propose. Moreover, the evaluation of our system is demonstrated in Section V. Finally, we draw the conclusion.

II. Related Work and Motivation

A. Blockchain Scalability Optimization

Since Nakomoto *et al.* [1] conceptualized the idea of Bitcoin, blockchain has aroused great interest from multitudinous fields, including resource management [16], [17], software defined network (SDN) [18], edge computing [19], [20], [21], [22], etc. Unfortunately, excessive resource requirements result in insufficient system scalability, which limits the potential for blockchain deployment in high workload scenarios with power-constrained devices. To further expand the application range of blockchain, Ehmke *et al.* [23] developed a scalable blockchain protocol called Proof-of-Property (PoP). PoP allows for most devices to validate incoming transactions without downloading the whole blockchain initially. Dorri *et al.* [24] designed a private immutable ledger, which acts similarly to blockchain, but is managed centrally. Among all peers, resource-rich devices create an overlay network to implement a publicly accessible distributed system. Additionally, Li *et al.* [25] upgraded ledger structure from linearity to DAG. By recognizing the contributions of all forks, DAG-based protocols significantly increase the mining efficiency [26]. In RapidChain [31], Zamani *et al.* presented the committee-based sharding technology to divide the whole P2P network into smaller committees. Thus, the broadcast latency and storage consumption can be mitigated. Moreover, Bitcoin-NG [32] and Multichain [33] developed leadership selection and cross-chain mechanism to optimize the scalability of blockchain, respectively. However, most existing systems only performed optimization toward a single aspect, e.g., computing power utilization, or adopted centralized measures. In contrast, we conduct decentralized alleviation from multiple perspectives.

B. Challenges From Resource Constraints

IIoT devices mainly contain sensors, control facilities, industrial equipment, etc. Generally, their hashrates are at MH/s level. While mining machines, such as ANTMINER S9 Hydro, have reached up to 18 TH/s [27]. If consensus mechanism is not improved, the hashrates of IIoT devices cannot afford the demand for mining. Additionally, most IIoT devices have low power constraints, but the energy waste of mining is awful [28]. From the perspective of synchronize strategy, peers can be classified into full nodes and lightweight nodes. As aforementioned in Section 1, full nodes implement complete decentralization at the cost of saving all the transaction data and operation records of the whole P2P network. Take Bitcoin as an example, the total size of local ledger reaches 200.3 GB [29]. Moreover, considering that the average block capacity is 1.03 MB, the block generation interval is 600 s. Local ledger will grow at a rate of 148.32 MB per day without recycling mechanism. Therefore, we summarize that the storage resources of IIoT devices cannot meet the requirements for storing numerous data

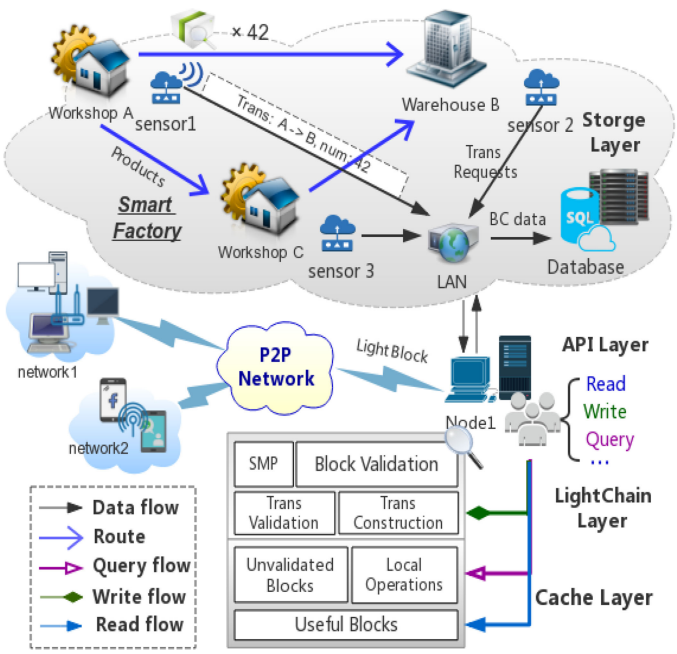

Fig. 1. Four-layer framework of LightChain.

of blockchain network. In terms of network resources, IIoT scenario is characterized by large scale and dynamic heterogeneity [30]. Meanwhile, the loss of packets and delay may also have an impact on network performance. Overall, the resources of IIoT devices are insufficient to deploy blockchain systems. Motivated by such fact, we develop a lightweight blockchain system to make blockchain open in IIoT.

III. System Model

In this section, we demonstrate the framework of LightChain in detail. From top to bottom, our blockchain system consists of four layers, i.e., *API layer*, *LightChain layer*, *Cache layer*, and *Storage layer* (as shown in Fig. 1).

A. LightChain Framework

We deploy LightChain in a smart factory to illustrate the framework of our blockchain system. As shown in Fig. 1, the factory is composed of two workshops A, C, and a warehouse B. Each department has Radio Frequency Identification (RFID) sensors to detect the trend of goods. Suppose that 42 products are processed in A and transported to B for storage, sensor1 will issue a transaction request T_r^{json}: *["from": A, "to": B, "amount": 42]*. Through Local Area Network (LAN), T_r^{json} is transmitted to the duty node Node1, whose API layer sets the action type as *trans-write* and sends it along with T_r^{json} to LightChain layer.

In LightChain layer, we design multiple validation mechanisms to confirm the validity and integrity of pending T_r, i.e., *digital signature validation* and *attachment validation*. The former executes a Pay to Public Key Hash script [36] to defend against double-spending attacks and ensure that T_r is not tampered during propagation. Since each transaction is allowed to pack a binary attachment, *Attachment validation* mainly verifies whether the type and format of T_r *attachment* are standardized. T_r will enter local transpool and wait to be finally packaged into the latest block, only if it passes all validations. To implement data traceability, industrial records will be preserved in parallel among geographically distributed devices.

Constrained by the huge size of blockchain ledger and the limited storage capacity of control nodes, we divide data storage into two parallel strategies: 1) *Dynamic storage:* Node1 only caches useful data filtered by UBOF. This part can be used as transaction verification or directly referenced by subsequent transactions. *2) static storage:* The complete copy of blockchain ledger and the integral list of operation records are stored in the cloud database of this production department as the backup source.

B. Layer Functions

API layer offers interfaces for industrial control applications, which abstract the functions of Cache layer and LightChain layer to provide various calls. Specifically, the API layer contains the following operations:

1) Action types (*block-read*, *trans-write*, etc.) abstract a set of operations and provide corresponding interfaces for clients to manage local ledger, check information or communicate with peers.
2) Policy configuration is designed to set the operation permission to local data for other IIoT devices.
3) Query operation can query operation record of other IIoT devices on local data, where the latest operations of local data are stored in Cache layer.

LightChain layer integrates multiple modules of blockchain, whose function will be elaborated in Section IV.

Cache layer is designed to accelerate the responses to various calls. The data cache here principally contains local operations, pending blocks, and useful blocks. Such content will be managed by LightChain layer.

Storage layer, usually served by resource-rich devices, provides persistent storage service for the upper layers. Note that since full nodes perform transaction validations merely by using the data in Cache layer, the "devices" here are not limited to blockchain participants.

IV. LightChain Layer Design

A. Architecture Overview

In the four-layer framework of LightChain, LightChain layer is the core level, the logic and function of blockchain are reflected here. Adopting modular design, LightChain layer is divided into three collaborating modules, i.e., *Webchain*, *Conchain*, and *Chainbase* (as shown in Fig. 2). Among them, Conchain/Webchain with Chainbase forms a Client/Server (C/S) structure. As shown in Table I, we define various message types on client sides, triggering the corresponding functions in Chainbase to complete basic operations of blockchain, such as UB filtering, transaction verification, etc. Each module communicates with others through Local Socket. In Conchain, we present

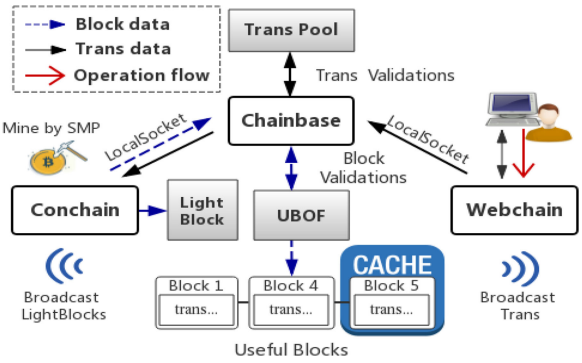

Fig. 2. Architecture of LightChain layer.

TABLE I
DEFINITIONS OF MESSAGE TYPES

Message Types	Attachment	Invoker
MsgType.Type_Block_Write	pending B_i^{bin}	Conchain
MsgType.Type_Trans_Write	pending T_i^{bin}	Webchain
MsgType.Type_Trans_Retrieve	missing T_i^{bin}	Conchain
MsgType.Type_UB_Filtering	$\beta = 0$ *or* 1	Chainbase
MsgType.Type_Response_Ok	None	Con/Webchain
MsgType.Type_Response_Error	Error type	Con/Webchain

a green consensus algorithm called SMP to reduce the computational power usage of IIoT devices. Additionally, due to the information overlap during block broadcast, we propose a lightweight data structure named LB for instead. In Chainbase, we design a novel UBOF to filter and offload insignificant data in cache layer. We will give an elaborate description about these modules in a block-cycle order, which indicates the process from submitting a transaction to saving it in blockchain.

B. Webchain

Webchain is the frontend of LightChain layer, which converts action types and JSON flow sent by API layer into predefined message types and binary flow processed internally. When receiving T_r^{json}, Webchain performs deserialization, extracts T_i*_Inputs*, T_i*_Outputs*, and constructs a corresponding transaction object T_i. Note that LightChain adopts bitcoin-based incentives; thus, transactions consist of *coinbase transactions* and *normal transactions*. For information integrity, security component will make a SHA256 digest of T_i, then encrypt it using Elliptic Curve Digital Signature Algorithm. As digital signature is completed, the binary flow T_i^{bin} will be sent to Chainbase for further operations.

C. Conchain

As the name suggests, Conchain acts as the consensus and P2P network module. Apart from executing consensus calculation based on SMP, it is responsible for managing peer list and establishing TCP/IP connections with them. After localhost digs out a new block or receives a block verification request, Conchain sends the message *MsgType.Type_Block_Write* appended with B_i^{bin} (the binary flow of pending block) to Chainbase.

1) Green Consensus Mechanism—Synergistic Multiple Proof: Blockchain is a ledger jointly maintained by distributed consensus operation and storage. Currently, it has two types: public chain and consortium (private) chain. LightChain is open to all IIoT nodes and completes transactions in a nontrusted environment; thus, belongs to public chain. In most cases, public chain uses PoW for parallel consensus, such as Bitcoin and Ethereum. PoW means setting workload as the basis for competition, in which miners constantly perform hash collisions unless its result is less than a predefined target. Since traditional PoW simply competes for computing power, target continues to decline and demands massive computational power to support. The process of PoW is defined as follows:

Find nonce

$$\text{SHA256}\left(\text{SHA256}\left(\text{block} + \text{nonce}\right)\right) < \text{target} \tag{1}$$

where "+" is the string concatenate operator. The smaller the target is, the more difficult (difficulty is defined by $\frac{\text{target}_{\max}}{\text{target}}$) the mining is.

Observing the limited computational resources of IIoT devices, we propose a green consensus mechanism called SMP. Different from previous work, such as Proof-of-Stake (PoS), Delegated PoS (DPoS), etc., we intend to promote collaboration among numerous IIoT devices. Apart from workload competition, SMP also introduces the competition for the degree of collaboration. Collaboration degree is related to the contributions of miners in ledger maintenance. We first present an indicator called Collaboration Index ($\mathcal{CI}$) for visualizing the collaboration of each peer node.

In our framework, $\mathcal{CI}$ can be gained by two approaches: first, system will reward block proposers a certain number of $\mathcal{CI}$ to stimulate their enthusiasm of mining and second, second, block proposers can earn $\mathcal{CI}$ from the transactions carried by self-proposed blocks. Correspondingly, the data flow between IIoT nodes recorded by transactions, i.e., collaborations, should be paid using $\mathcal{CI}$. The rate of $\mathcal{CI}$ is called collaboration fee $\mathcal{F}$. It determines the $\mathcal{CI}$ cost of transaction applicants. $\mathcal{F}$ is dynamically determined by

$$\mathcal{F} = \frac{\psi'}{\psi \times n}\, \mathcal{CI}/kB. \tag{2}$$

ψ: predefined throughput threshold;
ψ': average throughput of the entire network during recent 200 block-cycles;
n: the number of participants in P2P network.

Note that ψ represents the capacity of P2P network. Similar to Tahoe [37] for TCP congestion control, LightChain network measures ψ based on the relationship between the transaction application rate and that of transaction processing. For precisely, if the number of pending transactions in local transaction pool tends to increase, we conclude that P2P network has reached the threshold, otherwise it still has redundancy. To ensure the validity of $\mathcal{F}$, ψ is adjusted at regular interval among the entire P2P network. Similar to IOTA [41], seed nodes in LightChain act as coordinators simultaneously to ensure the efficiently consensus on ψ and n. While as the scale of P2P network grows to a

certain degree, devices can exchange information with numerous dynamically updated neighbors. In this case, the coordinators can be taken out of service. According to (2), $\mathcal{F}$ will fluctuate with ψ', whose trend is positively correlated. If the real-time throughput ψ' is higher than ψ, which means that the load of P2P network exceeded the maximum value of network capacity. $\mathcal{F}$ will increase to raise the $\mathcal{CI}$ consumption of transaction applications, and the effect of reducing the system burden is achieved. On the contrary, $\mathcal{F}$ will decrease to stimulate transaction applications when ψ' is lower than ψ. The larger the amount of IIoT nodes is, the lower $\mathcal{F}$ will be in this framework.

Based on the $\mathcal{CI}$ framework, SMP implements individually dynamic difficulty. Moreover, we also set two adjustment rules as auxiliary to further ensure mining stability and avoid concentration of computing power. The hash formula for SMP is as follows:

Find nonce

$$\text{SHA256}(\text{SHA256}(\text{block} + \text{nonce})) < (\Omega + \Phi) \times \text{target} \quad (3)$$

$$\Omega = \sum_{i=1}^{\mathcal{E}} (m\mathcal{CI}_i \times \Delta t_i) \quad (4)$$

$$\Phi = \frac{\theta}{200} \times \frac{\psi'}{\psi} = \frac{\theta}{200} \times \mathcal{F} \times n. \quad (5)$$

Δt_i: the corresponding time LocalHost holds m$\mathcal{CI}_i$;
$\mathcal{E}$: the amount of m$\mathcal{CI}$ that localhost possesses;
θ: the number of self-proposed blocks during 200 block-cycles.

The minimum unit of $\mathcal{CI}$ is defined as m$\mathcal{CI}$ ($\mathcal{CI} \times 10^{-3}$) stored in a First Input First Output (FIFO) queue $\mathbb{Q} = [(m\mathcal{CI}_i, \Delta t_i), (m\mathcal{CI}_{i-1}, \Delta t_{i-1}), \ldots, (m\mathcal{CI}_1, \Delta t_1)]$. As shown in (3), SMP adopts multiple difficulty assignments, where Ω abstracts the historical rights accumulated by one node since it participated in collaboration. Similarly, Φ represents the direct contributions miners made in recent cooperation. Although miners can only obtain $\mathcal{CI}$ after creating new blocks, continuous mining participation is also recognized as collaborating contributions. Since PoW-based protocols require wasted power to defend against double-spending attacks, we propose Δt_i, which is defined as the corresponding time local host holds m$\mathcal{CI}_i$. Δt_i is also included in the difficulty metric of SMP. Therefore, regardless of the result of mining competition, the workload of all participants is regarded as contributions for the cooperative ledger maintenance. Considering that each node holds different $\mathcal{CI}$ and Δt_i, the mining difficulty under SMP is independent, which ensures the fairness among participants. In most cases, $(\Omega + \Phi) > 1$ is satisfied; thus, the computational power required for miners is significantly reduced. Recall that the computational consumption of PoW is $\mathbb{E}_{\text{PoW}} = \frac{\text{target}_{\max}}{\text{target}} \times 2^{32}$, SMP reduces it to $\mathbb{E}_{\text{SMP}} = \frac{\text{target}_{\max}}{\Omega \times \Phi \times \text{target}} \times 2^{32} = \frac{1}{\Omega \times \Phi} \mathbb{E}_{\text{PoW}}$. However, for poor nodes with only a few $\mathcal{CI}$, they still cannot calculate the qualified nonce quickly. Therefore, we design two adjustment rules.

1) *Rule 1* (*Winner Initialization):* This rule requires that in each round of mining competitions, the participant who proposes the new block to clear its Δt_i (*for* $i \in (1, \mathcal{E})$). However, losers are allowed to maintain current Δt_i and continue to accumulate. Over time, consecutively failed participants gradually accumulate Δt_i and have an increasing chance to obtain block creation rights.
2) *Rule 2* (*Rotational Competition):* The winner have to wait for several blocks (*Lower bound* $\mathcal{L}$) to join the next competition. Assuming that $\mathcal{L} = \delta$, the number of competitors will drop from n to $n - \delta$, reducing the insignificant computational resources consumption of δ participants. Meanwhile, this rule also limits the longest free waiting gap to prevent rich participants from withhold $\mathcal{CI}$ and not participating in competition (*Upper bound* $\mathcal{R}$). Suppose that one node has been waiting for $\mathcal{R} + r$ before rejoining collaboration, the total amount of $\mathcal{CI}$ received by its first self-proposed block will be reduced from $\Gamma(0)$ to $\Gamma(r)$:

$$\Gamma(r) = \frac{\Gamma(0)}{\mathrm{r}} \times \frac{\psi}{\psi + \psi'}. \quad (6)$$

The main purpose of these constraints is to achieve the stability of distributed consistency computing, which involves two aspects: *first, the degree of decentralization throughout P2P network* and second, *the stability of block generation rate and mining participation*. Through forced adjustments, *Rule 1* and $\mathcal{L}$ significantly mitigate the influence of Matthew Effect [39]. Simultaneously, the design of $\mathcal{R}$ and $\mathcal{F}$ maintains the mining enthusiasm. Furthermore, the wasted energy of mining competition is significantly declined due to the constraint on miner scale ($\mathcal{L}$). Combining multiple proof with adjustment rules, we alleviate the computing power consumption of blockchain system as aforementioned in Section I. Moreover, we upgrade traditional PoW into a novel green consensus mechanism. In SMP, miners hold high mining efficiency without affecting network security, which we will demonstrate in Section V-C.

2) Information Overlap Mitigation: Generally, the content of block broadcast is: *block_header* + *trans_list*,

1) block_header: the list of basic parameters, including *hash, prev_hash, nonce, timestamp*, root of *Merkle Hash Tree (MHT)*, etc.
2) trans_list: the list of all transactions belong to this block, each item contains *transaction_inputs/outputs, timestamp, txid*, *version, signature, public_key*, etc.

Among them, *txid* is defined as follows and can uniquely identify a transaction:

$$\text{txid} = \text{SHA256}(\,\text{timestamp}^{\text{bin}} + \text{version}^{\text{bin}} + \text{signature} + \text{in/outputs}^{\text{bin}} + \text{public\_key}\,). \quad (7)$$

We observe that full nodes broadcast the complete transaction information once to peers straightway after receiving transaction requests from clients. Then, such transactions will be packaged in pending blocks and broadcast twice, causing the information overlap and not conducive to improving throughput.

In P2P network, nodes that receive new message will participate in broadcasting immediately, which means the number of informed nodes is exponentially increased. Therefore, pending transactions can be accurately and rapidly propagated to the entire network in limited time. This mechanism guarantees a high level of data consistency between peer nodes, i.e., for the transpool of sender, denoted as $\mathbb{T}_S$ and that of receiver, denoted

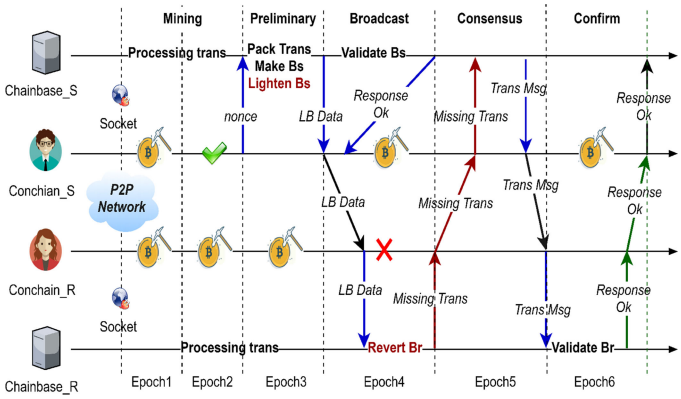

Fig. 3. Process of broadcasting and receiving LB.

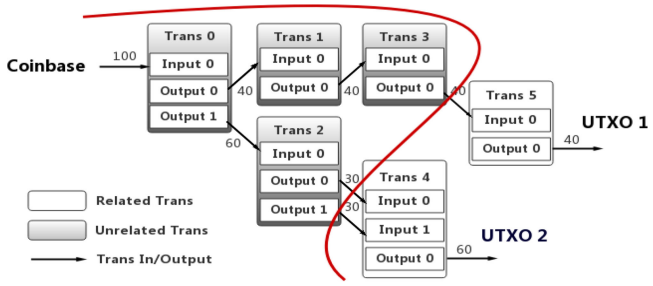

Fig. 4. Graph structure of transactions.

as $\mathbb{T}_R$, the following condition is satisfied:

$$\forall T_i \in \mathbb{T}_S, \text{Prob}\left(T_i \in \mathbb{T}_R\right) \approx 1. \tag{8}$$

Considering this situation and utilizing *txid*'s property of uniqueness as hash output, we define a lightweight data structure named LB to avoid secondary broadcast of transactions. The six epochs of sending and receiving LB is shown as Fig. 3.

As sender generates the basic block B_S, a LB instance *LB* is constructed by the constructor *LightBlock* (B_S). *LB* also has two components: *block_header* and *txid_list*, where *block_header* is the same as B_S.*block_header*, *LB.txid_list* $= \{T_0.txid, T_1.txid, \ldots, T_{n-1}.txid, T_n.txid\}$ (lines 2–3 in Algorithm 1). For the receiver, it finds matched transactions using *LB.txid_list* (lines 9–10 in Algorithm 1) to fill *txid_list*. Combining B_S.*block_header* with *txid_list*, B_R shares integrated unity with B_S while streamlines the broadcast content. In Fig. 3, T_n is not matched, then receiver will send *MsgType.Type_Trans_Retrieve* to apply for the entire content of the missing T_n (lines 12–15 in Algorithm 1). Considering that the size of *txid* is a fixed value of 32 B, while one complete transaction occupies more than 250 B, broadcasting LBs instead of blocks can effectively decrease the degree of data overlap. Since peers synchronize data in parallel, overlapped part makes up a large proportion of all broadcast information. Therefore, our proposal can significantly reduce the network usage and contributes to alleviate the problem of insufficient bandwidth in IIoT scenarios. Meanwhile, the propagation latency of LB is not sensitive to network conditions, possessing better scalability for network heterogeneity.

Algorithm 1: LightBlock Operations.

```
1:  procedure BLOCK-LIGHTWEIGHTING
2:      for all T_i ∈ B_S.trans_list do
3:          LB.txid_list = LB.txid_list ∪ {T_i.txid};
4:      end for
5:  end procedure
6:
7:  procedure LB-REVERSION
8:      for all txid ∈ LB.txid_list do
9:          if txid == T_j.txid (0 ≤ j ≤ n) in T_R then
10:             B_R.trans_list = B_R.trans_list ∪ {T_j};
11:         else
12:             return MsgType.Type_Trans_Retrieve;
13:         end if
14:     end for
15:     return MsgType.Type_Response_Ok;
16: end procedure
```

D. Chainbase

Chainbase is the backend of LightChain layer, owning the permission to add, filter, or offload local ledger data cached by cache layer. For example, when receiving *MsgType.Type_Block_Write* from Conchain, the constructor of block will be triggered, whose input is B_i^{bin}. Here, deserialization code convert B_i^{bin} into a pending block object B_i. If $B_i.prev\_hash = B_{i-1}.hash$ and SHA256 (SHA256 $(B_i^{\text{bin}} + B_i.nonce)) < target$ are satisfied, Chainbase will return *MsgType.Type_Response_Ok* to declare that one miner has approved the pending block. After B_i acquires confirmations by a certain number of miners across P2P network, it becomes a legal block. Additionally, Chainbase is responsible for managing local transpool, verifying transactions, etc.

1) Transaction validations in UTXO-based blockchain: Transactions in blockchain are mainly based on the UTXO mechanism [40]. In the inputs field, the subsequent transaction references a list of outputs belonging to one or more previous transactions, and indicates the indexes of outputs in transactions where they are packaged. Through backtracking to genesis block, full nodes form the graph structure of transactions and establish a complete UTXO database (as shown in Fig. 4). If an output referenced by the pending transaction does not exist in UTXO, it will be judged as double-spending. That is to say, the referenced outputs (the gray part of Fig. 4) cannot be used as a basis for proving the validity of a subsequent transaction, let along being directly referenced again.

2) Ledger Data Offloading Theory: UTXO-based verification process enlightens us that for the content of dynamic storage, we can only save data that might be directly referenced by subsequent transactions. As to the insignificant part, it ought to be offloaded from local without influence. Therefore, we present a novel offloading algorithm named UBOF aiming at filtering unrelated blocks. We will give definitions about some concepts.

Definition 1. (Transaction_Input): In LightChain, the input form of transaction T_i is *(txid, output_index)* illustrating that this input references transaction output $T_{i-k}\_Output_{output\_index}$ of T_{i-k} $(0 < k < i$, and $T_{i-k}.txid$ is equal to *txid*).

Algorithm 2: UBOF.

Input: B_n; $\mathbb{U} = \varnothing$; $\mathbb{B}$
Output: $\beta \in \{0, 1\}$ ($\mathcal{UB}$s in $\mathbb{U}$ are removed or not)

```
1:  procedure Chain-Data-Filter(B_n)
2:      β = 1;
3:      for all T_i ∈ B_n.trans_list do
4:          for all T_i_Input_j ∈ T_i do
5:              utxo_i = UTXOTable.Delete (T_i_Input_j);
6:              UTXONum[utxo_i.value [1] ] -= 1;
7:              if UTXONum[utxo_i.value [1] ] == 0 then
8:                  β = 0;
9:                  U = U ∪ {utxo_i.value[1]};
10:             end if
11:         end for
12:         for all T_i_Output_j ∈ T_i do
13:             UTXOTable.Add (T_i_Output_j, B_n.hash);
14:             UTXONum[B_n.hash] += 1;
15:         end for
16:     end for
17: end procedure
18:
19: procedureUB-Offloading U, β
20:     if β == 0 then
21:         for all B_i ∈ B do
22:             if B_i.hash ∈ U then
23:                 B = B \ {B_i};
24:                 U = U \ {B_i.hash};
25:             end if
26:         end for
27:         if U is ∅ then
28:             β = 1;
29:         end if
30:     end if
31:     return β;
32: end procedure
```

Definition 2. (Transaction_Output): Transaction_output dict {*"amount": Asset, "to": public_key_hash*} represents that the amount of products transferred to the destination address "*public_key_hash*" is "*Asset*".

Definition 3. (Unrelated Object): When an object only saves the function of tracing for self-examination or external query request, while does not have the ability to be directly referenced by subsequent transactions, we define it as unrelated object.

Definition 4. (UTXOTable): UTXOTable is a dict in the form of {(*txid, output_index*): ({*"amount": Asset, "to": public_key_hash*}, *block_hash*)}, whose *key* is the location of an UTXO, indicating the transaction it belongs to. The *value* is designed as a 2-element tuple, *value [0]* shows the detailed information of the output. Additionally, *value [1]* indicates the block hash where this UTXO is packaged.

Definition 5. (UTXONum): UTXONum ({*"block_hash": num*}) follows the key-value form and is designed to record the number of remaining UTXOs in each block.

The main operations performed by UBOF are to detect and delete unrelated blocks ($\mathcal{UB}$). According to the *Definition 3*, $\mathcal{UB}$ represents the block whose transaction outputs are all referenced, i.e., UTXONum [$\mathcal{UB}.hash$] $= 0$.

We describe UBOF in Algorithm 2. During each block-cycle, UBOF starts immediately after Chainbase extends the blockchain $\mathbb{B} = \{B_1, B_2, \ldots, B_{n-1}\}$. More explicitly, data filter first traverse all the transactions packaged by B_n, adding, or deleting entries in UTXOTable (lines 3–5, 12–13 in Algorithm 2). Meanwhile, we set two passive operations in UTXONum: *subjoin* and *subtract*, which are respectively triggered by the synonymous executions of UTXOTable (line 6, 14 in Algorithm 2). When the UTXONum of a certain block becomes 0, it will be filtered and appended to $\mathbb{U}$ (lines 7–9 in Algorithm 2). According to the indication of $\mathbb{U}$, $\mathcal{UB}$s are offloaded in the second step. Finally, UBOF returns a flag β. If $\beta = 1$, the above processes are judged valid, otherwise Chainbase reexecutes UBOF. By filtering and recycling stale data ($\mathcal{UB}$), the growth rate of ledger size gets constrained. Without UBOF, the greater the network throughput, the faster the consumption of storage room, i.e., the blockchain scalability optimization and the mitigation of resource occupation are contradictory. However, in LightChain, high workload not only brings numerous pending transactions. Based on UTXO mechanism, unrelated transactions are simultaneously increased, allowing UBOF to filter more stale data. Therefore, UBOF guarantees the rational storage resource occupation under high throughput, which can solve the problem of insufficient storage resources as we mentioned in Section I.

V. Evaluation and Analysis

A. Security Analysis

In this section, we analyze the defensive capability of SMP facing various types of attacks.

1) Double-Spending Attacks: Previous work indicates that the security of PoW-based protocols is attributed to the "wasted" computation [38]. Recall that SMP saves the wasted computation by constraining the scale of competitors ($\mathcal{L}$ in *Rule 2*). Intuitively, $\mathcal{L}$ may have negative impacts on network security. However, the scale and mining power of Byzantine attackers are limited by *Rule 2*, simultaneously. We consider that during each round of mining competition under SMP, the ratio of mining power mastered by attackers is approximately similar to that under PoW, except extreme cases where numerous distributed attackers constitute one interconnected entity. Furthermore, SMP adopts multiple metrics; thus, attackers need to lead in both $\mathcal{CI}$ and mining power to successfully initiate a double-spending attack. Therefore, the security level of PoW and SMP under double-spending attacks can generally be the same.

2) Nothing-at-Stake Attacks: In traditional PoW, miners generally mine atop only one chain when there exists parallel forks. Otherwise, it will dilute the valuable mining power. However, for PoS-based systems, rich nodes can recognize all pending chains simultaneously without any effect. Thus, they can maximize their benefits in any case, called Nothing-at-Stake

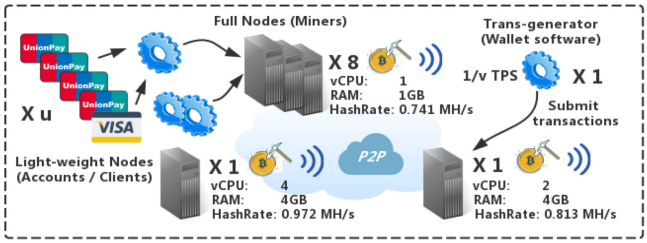

Fig. 5. Settings of evaluations.

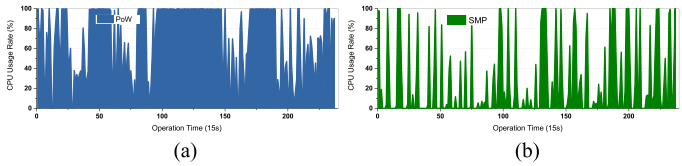

Fig. 6. CPU usage rate of mining process. (a) CPU usage rate under PoW, (b) CPU usage rate under SMP.

attacks. Under SMP, nodes still need computing power to propose blocks. Furthermore, $\mathcal{L}$ and Δt_i limit the frequency with which nodes participate in collaboration and the probability they win the competition. Therefore, peer nodes are hardly to initiate Nothing-at-Stake attacks in SMP-based blockchain networks, and the same in the case of Long range attacks.

B. Evaluation Setting

Testbed: We build a P2P network with ten elastic compute services (ECS) powered by aliyun. Each server is equipped with one Intel Xeon 2.5 GHz CPU, and runs Ubuntu 16.04.2 LTS with GUN/Linux 4.8.0-36-generic kernel. ECSs work as full nodes (miners) of LightChain. Note that, we simulate the practical conditions of IIoT when building our experimental network. As aforementioned in Section II, we mainly consider and implement the following characteristics of IIoT scenarios:

1) *Heterogeneity and geographical distribution:* To construct a heterogeneous P2P network, we configure the miners with different vCPUs and memory (as shown in Fig. 5). Moreover, for network distribution, we deploy ECSs in various regions of Asia (Shanghai-4, Hong Kong-3, Tokyo-3).
2) *Constrained resources:* All miners use CPU to perform the mining process, whose computing power is at MH/s level (as shown in Fig. 5), the same as IIoT devices. Additionally, the storage resources are also fixedly allocated (40 GB) without allowing for the unrestricted growth of local ledger.
3) *Complex network topology:* In our experiment, each ECS processes dedicated public IP address with a bandwidth limit of 1 Mb/s. Simultaneously, hosts in the same region construct independent LANs (1 Gb/s).
   Furthermore, peer nodes can maintain 1–9 incoming connections, which supports dynamic updates.

Simulator: On the basis of importing *cryptography [34]* and generating public–private key pairs with elliptic curve *ec.SECP256K1*, we develop a automatic trans-generator in Python 3. As an abstract wallet software, it controls several accounts and submits random transactions at regular interval. Here, accounts act as light-weight nodes/clients in P2P network.

Workload: In order to conduct comparative experiments under different workloads, we connect each miner to one generator and set two parameters: μ, ν. Among them, μ represents the number of accounts controlled by each generator, which is designed to adjust the scale of clients across the P2P network. ν is the time interval between two transactions be sent. Consequently, the efficiency of transaction generator is defined as the number of transactions initiated per second, denoted as $\mathrm{O}(c) = 1/\nu$. Moreover, considering the difference in user activity, we divide these accounts into primary and general ones. Primary accounts have a higher frequency of applying transactions.

C. Result Analysis

We evaluate LightChain to show the performance of following three inspections:

1) Inspection of SMP: CPU Usage Rate. First, we deploy Linux Agent to monitor the CPU usage rate of mining process. We develop traditional PoW, which is compared with our green consensus mechanism. The contrast between Fig. 6(a) and (b) illustrates that SMP can effectively reduce CPU workload and alleviate the impact on device's performance when deploying blockchain.

Block Generation Speed: The average block generation interval (abbreviated as interval) is 3.41 s under SMP, whose improvements are 3.86 × that of PoW (13.15 s), and 176.47 × that of Bitcoin (600 s). Here gives the maximum throughput of blockchain system

$$\mathsf{max}(\text{throughput}) = \frac{\text{block size}}{\text{trans size} \times \text{ interval}}. \tag{9}$$

Note that the premise of (9) is that the processing efficiency of blockchain system is greater than that of transaction generators, i.e., there are sufficient transactions to be verified in local transaction pool. Otherwise, the throughput of network is limited by the total capacity of all transaction generators, defined as $\mathsf{Num}(\textit{trans-generator})\cdot\mathrm{O}(c)$.

In Bitcoin, block size is limited to 1 MB, and interval is stabilized at 600 s. Assuming that transaction size is 250 B, thus throughput can be calculated as 6.67 Transaction Per Second (TPS). Substituting LightChain's interval and block size (1.5 MB), its theoretical maximum throughput can reach 1759.53 TPS. Hence SMP is sufficient to handle massive industrial data. Note that the actual value is affected by network condition, latency, and other factors.

Computational Cost: Fig. 7(a) shows the computational resource cost of full nodes, whose metric is defined as the total number of hash operations that a signal miner tried during 50 block-cycles. Besides configuration discrepancies, we also set different initial difficulties for miners to further enhance the heterogeneity of network (as shown in table of Fig. 5). Through dynamic difficulty, mining becomes greatly easier. Regardless of hardware hashrate, SMP outperforms PoW by up to 60.68%

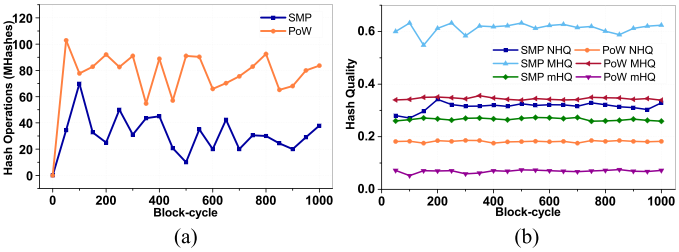

Fig. 7. Computational cost under PoW and SMP. (a) Comparison of hashes. (b) Comparison of HQ.

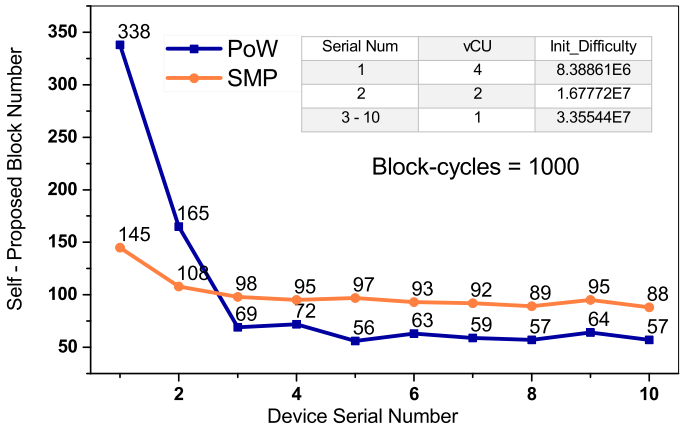

Fig. 8. Comparison of computational decentralization in PoW- and SMP-based P2P network.

computational cost reduction. Additionally, we calculate the average number of hash operations miners perform to propose a block. In PoW, mining one block demands 7.71 MH, while SMP reduces it to 4.49 MH. That is to say, peers can alleviate power consumption without damaging their interests. Besides dynamic difficulty, this improvement is also attributed to the adjustment of network computing resources by *Rule 2*.

Hash quality (HQ) is defined to measure the efficiency of LightChain for power utilization. This indicator is calculated by $\mathrm{HQ} = \frac{\mathrm{hashes}_p}{\mathrm{hashes}_a}$, where hashes_a represents the computational cost of miners. hashes_p is such hash number they paid for successfully proposing new blocks. We evaluate Network HQ (NHQ), and Peer HQ (PHQ). NHQ represents the mean of all miners, while "devices" in PHQ refer to single nodes. Fig. 7(b) shows NHQ and the extremum of PHQs (maximum: MHQ, minimum: mHQ). Recall that in SMP, the scale of competitors is limited by $\mathcal{L}$; thus, power-constrained miners also maintain PHQ up to 0.25. While under PoW, regardless of mining power, the PHQ for every miner is generally lower than that under SMP. Moreover, NHQ declines to less than 0.2, which is not contributed to improve the mining activity of miners. The comparison of Fig. 7 illustrates that the waste of meaningless computing power in P2P network is obviously reduced and mining efficiency get increased in LightChain.

Consensus Stability: Furthermore, we measure the degree of decentralization across the P2P network to investigate whether our *Rules 1* and *2* achieve the goal of maintaining consensus stability or not. Through Fig. 8, we conclude that under PoW, devices with high hashrate grasp the block creation right. As to SMP, due to the adjustment of two rules, block generation is obviously more dispersed, and the level of decentralization gets significantly upgraded.

Based on the experimental data, we demonstrate the validity of proposed SMP. Attributing to $\mathcal{CI}$ framework, the contributions of miners are measured fairly, then form dynamic difficulty. Therefore, the computational cost of mining is significantly reduced [see Fig. 7(a)]. Additionally, the constraint of adjustment rules alleviates wasted energy in P2P network and implements an efficient greening consensus [NHQ in Fig. 7(b)]. More importantly, SMP enhances the collaboration between devices. By accumulating Δt, power-constrained nodes also acquire block generation rights, which enhances the degree of decentralization (see Fig. 8). In general, SMP declines the blockchain-to-node computing power occupation. Moreover, it ensures IIoT devices remain sufficient resources to perform industrial tasks (see Fig. 6).

2) Inspection of LB: Network Usage. In order to maximize the streamlining of data, we compile dedicated serialization code without using existing tools, such as Pickle [35]. We serialize the genesis block B_g of LightChain, which encapsulates only one coinbase transaction inside. The size of B_G^{bin} is 442 B, while its corresponding $\mathrm{LB}_G^{\mathrm{bin}}$ only occupies 119 B, reducing to 0.26 $\times$. On average, the process of lightening *Block* costs 10.52 ms (containing 1000 transactions), reverting *LB* costs 63.45 ms. Since these steps are only performed once per block-cycle, there is no impact on system efficiency. In the case of not triggering any exceptions, the latency of verifying a pending transaction is less than 0.95 ms.

We conduct four sets of evaluations via counting the total amount of data broadcast and received by one peer node, where system throughput increases from 100 to 1000 TPS, stepping 300 TPS. Since our experimental throughputs are less than the throughput threshold (1759.53 TPS) in the network environment configured as Fig. 5, then system workload can be adjusted by $\mathrm{O}(c)$. Recall that $throughput = \mathrm{Num}(trans\text{-}generator)\cdot\mathrm{O}(c)$. Considering $\mathrm{Num}(trans\text{-}generator) = 10$, we can calculate the values of ν in the above workload states are [0.1 s, 0.025 s, 0.015 s, 0.01 s] (shown in the vertical axis of Fig. 9), respectively. As shown in Fig. 9, using LB instead of entire block for broadcast can save up to 90.55% network usage, markedly reducing the network bandwidth requirements of blockchain. The greater the system throughput, the more effective our proposal is.

3) Inspection of UBOF: UB Filtering. In terms of UB filtering, we simulate multiple environments by adjusting μ, ν. Fig. 10(a) and (b), respectively, shows the number of remaining blocks filtered by UBOF under different scale of clients and throughput. The results demonstrate that UBOF works stably in all states of system, filtering 41.6–51.4% of ledger data.

Storage Cost: For more precise demonstration, we set ν to [0.1 s, 0.01 s] successively, so that throughput (in this case, system throughput is defined as $10\cdot\mathrm{O}(c)$.) equals to [100 TPS, 1000 TPS]. Such settings respectively indicate the idle and high workload environments of IIoT. By measuring the size of data

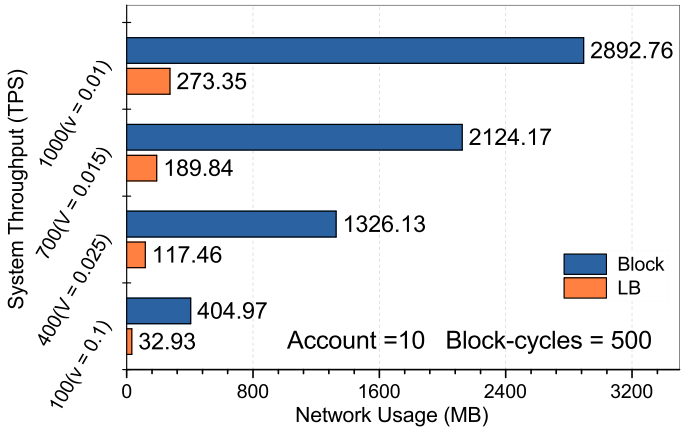

Fig. 9. Comparison of network usage for data synchronization.

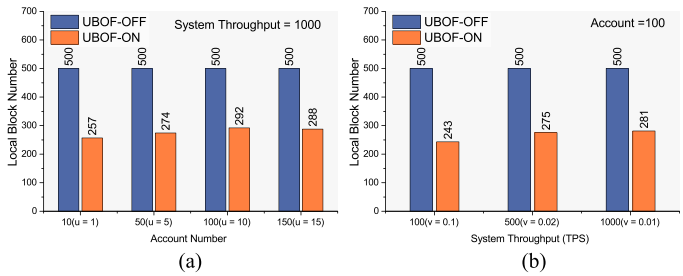

Fig. 10. Comparison of ledger data. (a) Remaining blocks in different scale of accounts. (b) Remaining blocks in different scale of throughput.

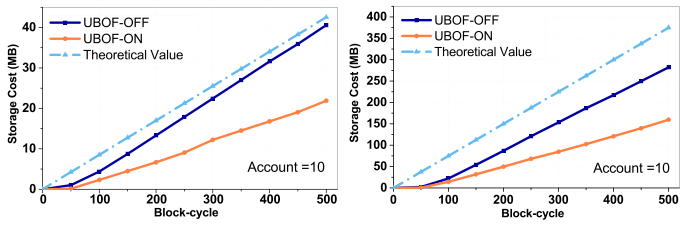

Fig. 11. Comparison of storage cost. (a) Throughput = 100 TPS. (b) Throughput = 1000 TPS.

saved in Cache layer, we compare the storage cost before and after running UBOF (as shown in Fig. 11). On average, UBOF can offload up to 43.35% of the storage resource consumption. With a throughput of 1000 TPS, the savings have reached 122.59 MB when only 500 blocks are mined. Moreover, the recycle rate can maintain at around 50% under different scale of system throughput, which contributes to deploy LightChain in high workload environments. Note that theoretical value $\bar{T}$ in Fig. 11 represents the theoretical scale of storage cost under current throughput, defined as

$$\bar{T} = \mathrm{Num}(block\text{-}cycle) \cdot (\text{trans size} \cdot 10 \cdot O(c)) \qquad (10)$$

where we approximate trans size = 250B, the same as in Bitcoin (10). As shown in Fig. 11, UBOF-OFF curve is closely parallel to theoretical value. In the first 100 block-cycles, UBOF-OFF is flatter due to the insufficient volume of pending transactions. Based on this insight, we conclude that Lightchain does process transactions at the set rate, which verifies the validity of transaction generator and authenticity of our experiments.

VI. Conclusion

To alleviate resource occupation of blockchain and make it suitable for IIoT scenarios, we propose a lightweight blockchain system (LightChain) in this paper. LightChain has the characteristics of resource-efficient without affecting the traceability and nonrepudiation of blockchain. Specifically, we propose a green consensus mechanism called SMP, to reduce the consumption of computing power. For information overlap optimization, we design a lightweight data structure named LB in broadcast. Moreover, we elaborate two parallel ways for storage and a novel UBOF filter to mitigate the burden of storage cost. Finally, extensive experiments show the advantages and superiority of our proposal.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," [Online]. Available: https://bitcoin.org/bitcoin.pdf, 2008.
[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
[3] J. Robert, S. Rauh, H. Lieske, and A. Heuberger, "IEEE 802.15 low power wide area network (LPWAN) PHY interference model," in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, USA, 2018, pp. 1–6.
[4] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.
[5] H. Li, K. Wang, T. Miyazaki, C. Xu, Y. Sun, and S. Guo, "Trust-enhanced content delivery in blockchain based information-centric networking," *IEEE Netw*, vol. PP, no. 99, pp. 1–7, May. 2019.
[6] "RuffChain," 2018. [Online]. Available: https://ruffchain.com
[7] B. Benshoof, A. Rosen, A. G. Bourgeois, and R. W. Harrison, "Distributed decentralized domain name service," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, Chicago, IL, USA, 2016, pp. 1279–1287.
[8] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
[9] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Inform.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.
[10] M. Hossain, R. Hasan, and S. Zawoad, "Probe-IoT: A public digital ledger based forensic investigation framework for IoT," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, 2018, pp. 1–2.
[11] K. Karlsson *et al.*, "Vegvisir: A partition-tolerant blockchain for the internet-of-things," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, Vienna, 2018, pp. 1150–1158.
[12] J. Ellul and G. J. Pace, "AlkylVM: A virtual machine for smart contract blockchain connected internet of things," in *Proc. 9th IFIP Int. Conf. New Technologies, Mobility Secur.*, Paris, Italy, 2018, pp. 1–4.
[13] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Joint computation offloading and content caching for wireless blockchain networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Honolulu, HI, USA, 2018, pp. 517–522.
[14] "Python 3." 2019. [Online]. Available: https://www.python.org
[15] "Gunicorn." 2017. [Online]. Available: https://gunicorn.org
[16] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proc. USENIX Conf. Usenix Annu. Tech. Conf.*, Denver, CO, USA, 2016, pp. 181–194.
[17] C. Xu, K. Wang, and M. Guo, "Intelligent resource management in blockchain based cloud data centers," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 50–59, Dec. 2017.
[18] P. K. Sharma, S. Singh, Y. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017.

[19] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Netw.*, vol. 32, no. 3, pp. 78–83, May/Jun. 2018.
[20] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things*, vol. PP, no. 99, pp. 1–12, Oct. 2018.
[21] C. Xu, K. Wang, P. Li, S. Guo, J. Luo, B. Ye, and M. Guo, "Making big data open in edges: A resource-efficient blockchain-based approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 4, pp. 870–882, Apr. 2019.
[22] C. Xu, K. Wang, G. Xu, P. Li, S. Guo, and J. Luo, "Making big data open in collaborative edges: A blockchain-based framework with reduced resource requirements," in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, USA, 2018, pp. 1–6.
[23] C. Ehmke, F. Wessling, and C. M. Friedrich, "Proof-of-property—A lightweight and scalable blockchain protocol," in *Proc. IEEE/ACM 1st Int. Workshop Emerg. Trends Softw. Eng. Blockchain*, Gothenburg, Sweden, 2018, pp. 48–51.
[24] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized BlockChain for IoT," in *Proc. IEEE/ACM 2nd Int. Conf. Internet-of-Things Des. Implementation*, Pittsburgh, PA, USA, 2017, pp. 173–178.
[25] C. Li, P. Li, D. Zhou,W. X. F. Long, and A. C. Yao, "Scaling Nakamoto consensus to thousands of transactions per second," 2018. [Online]. Available: https://arxiv.org/abs/1805.03870
[26] F. M. Beni and I. Podnar arko, "Distributed ledger technology: Blockchain compared to directed acyclic graph," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, Vienna, 2018, pp. 1569–1570.
[27] "ANTMINER S9 Hydro miner." 2018. [Online]. Available: https://www.bitmain.com
[28] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.
[29] "Ledger size of Bitcoin," 2019. [Online]. Available: https://www.blockchain.com/zh-cn/charts
[30] I. Bedhief, M. Kassar, and T. Aguili, "SDN-based architecture challenging the IoT heterogeneity," in *Proc. 3rd Smart Cloud Netw. Syst.*, Dubai, 2016, pp. 1–3.
[31] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, pp. 931–948.
[32] I. Eyal, A. E. Gencer, and R. V. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Netw. Syst. Des. Implementation*, 2016, pp. 45–59.
[33] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, Vienna, Austria, 2018, pp. 1203–1211.
[34] "Cryptography," 2019. [Online]. Available: https://pypi.org/project/cryptography/
[35] "Pickle," 2018. [Online]. Available: https://docs.python.org/3/library/pickle.html
[36] "Pay-to-Public-Key-Hash (P2PKH)," 2008. [Online]. Available: https://bitcoin.org/en/glossary/p2pkhaddress
[37] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 959–971, Dec. 2003.
[38] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2016, pp. 3–16.
[39] M. Yang, Y. Zhou, Q. Zhou, K. Chen, J. He, and X. Yang, "Observation of Matthew effects in Sina Weibo microblogger," in *Proc. IEEE Int. Conf. Big Data*, Silicon Valley, CA, USA, 2013, pp. 41–43.
[40] "Unspent Transaction Output (UTXO)," 2008. [Online]. Available: https://bitcoin.org/en/glossary/unspent-transaction-output
[41] "IOTA," 2016. [Online]. Available: https://www.iota.org/

Yinqiu Liu received the Bachelor degree in internet of things engineering. He is working toward the undergraduate degree in the College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China.

His current research interests include big data, distributed and parallel computing, and blockchain applications and optimization.

Kun Wang (M'13–SM'17) received two Ph.D. degrees, from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2009 and from the University of Aizu, Aizuwakamatsu, Japan, in 2018, both in computer science.

He was a Postdoctoral Fellow at the University of California, Los Angeles (UCLA), CA, USA, from 2013 to 2015, and a Research Fellow at the Hong Kong Polytechnic University, Hong Kong, from 2017 to 2018. He is currently a Research Fellow at UCLA. His current research interests include big data, wireless communications and networking, energy Internet, and information security technologies.

Dr. Wang was the recipient of the Best Paper Award at IEEE GLOBECOM16. He serves as an Associate Editor of IEEE ACCESS, an Editor of *Journal of Network and Computer Applications,* and Guest Editor of IEEE NETWORK, IEEE ACCESS, *Future Generation Computer Systems, Peer-to-Peer Networking and Applications*, and *Journal of Internet Technology*.

Yun Lin received the B.S. degree in computer science from Dalian Maritime University, Dalian, China, in 2003, the M.S. degree from Harbin Institute of Technology, Harbin, China, in 2005, and the Ph.D. degree from Harbin Engineering University, Harbin, China, in 2010.

He was a Visiting Scholar at Wright State University, Dayton, OH, USA, from 2014 to 2015. He is currently an Associate Professor with Harbin Engineering University, and his research interest focuses on communication technology, signal processing, information fusion, cognitive radio, and software-defined radio.

Wenyao Xu (M'12–SM'18) received the Master's and Bachelor's degrees in computer science from Zhejiang University, Hangzhou, China, and the Ph.D. degree in electronic engineering from the University of California, Los Angeles, CA, USA.

He is an Associate Professor with tenure at the Computer Science and Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA. His group has focused on exploring novel sensing and computing technologies to build up innovative Internet-of-Things (IoT) systems for high-impact human-technology applications in the fields of Smart Health and Cyber-Security. Results have been published in peer-reviewed top research venues across multiple disciplines, including computer science conferences (e.g., ACM MobiCom, SenSys, MobiSys, UbiComp, ASPLOS, ISCA, HPCA and CCS), biomedical engineering journals (e.g., IEEE TBME, TBioCAS, and JBHI), and medicine journals (e.g., LANCET). To date, his group has published more than 160 papers, won six best paper awards, two best paper nominations, and three international best design awards. His inventions have been filed within U.S. and internationally as patents, and have been licensed to industrial players. His research has been reported in high-impact media outlets, including the Discovery Channel, CNN, NPR, and the *Wall Street Journal*.

Dr. Xu serves as an Associate Editor of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, and on technical program committees of numerous conferences in the field of Smart Health and Internet of Things, and was a TPC co-chair of IEEE Body Sensor Networks in 2018.