This is the Pre-Published Version.

The following publication L. Lao, X. Dai, B. Xiao and S. Guo, "G-PBFT: A Location-based and Scalable Consensus Protocol for IoT-Blockchain Applications," 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 2020, pp. 664-673 is available at https://doi.org/10.1109/IPDPS47924.2020.00074.

G-PBFT: A Location-based and Scalable Consensus Protocol for IoT-Blockchain Applications

Laphou Lao*, Xiaohai Dai*†, Bin Xiao* and Songtao Guo‡
*The Hong Kong Polytechnic University, Hong Kong,
†Huazhong University of Science and Technology, China,
‡Chongqing University, China

***Abstract*—IoT-blockchain applications have advantages of managing massive IoT devices, achieving advanced data security, and data credibility. However, there are still some challenges when deploying IoT applications on blockchain systems due to limited storage, power, and computing capability of IoT devices. Applying current consensus protocols to IoT applications may be vulnerable to Sybil node attacks or suffer from high-computational cost and low scalability. In this paper, we propose G-PBFT (Geographic-PBFT), a new location-based and scalable consensus protocol designed for IoT-blockchain applications. The principle of G-PBFT is based on the fact that most IoT-blockchain applications rely on fixed IoT devices for data collection and processing. Fixed IoT devices have more computational power than other mobile IoT devices, e.g., mobile phones and sensors, and are less likely to become malicious nodes. G-PBFT exploits geographic information of fixed IoT devices to reach consensus, thus avoiding Sybil attacks. In G-PBFT, we select those fixed, loyal, and capable nodes as endorsers, reducing the overhead for validating and recording transactions. As a result, G-PBFT achieves high consensus efficiency and low traffic intensity. Moreover, G-PBFT uses a new era switch mechanism to handle the dynamics of the IoT network. To evaluate our protocol, we conduct extensive experiments to compare the performance of G-PBFT against existing consensus protocol with over 200 participating nodes in a blockchain system. Experimental results demonstrate that G-PBFT significantly reduces consensus time, network overhead, and is scalable for IoT applications.**

***Index Terms*—IoT, blockchain, consensus protocol, PBFT, geographic location, scalable**

I. Introduction

Internet of Things (IoT) technology can connect massive devices to work collaboratively and automatically without manual intervention. IoT devices can generate different kinds of data to serve various purposes. By 2020, the number of IoT devices like smartphones, smart home appliances, and various types of sensors will increase to 20 billion [1]. However, there are many challenges in traditional IoT applications, including massive devices management, data integrity, and data robustness.

Blockchain eliminates the need of a central server and offers high data availability, security, transparency, and immutability for finance, IoTs, and other applications. Blockchain technology paves the way for addressing traditional IoT problems by providing IoT data integrity. The public blockchain is widely used in cryptocurrencies, e.g., Bitcoin and Ethereum. The private blockchains can be used in corporate business applications. In general, blockchain is considered to be able to establish a trust relationship among unauthentic entities.

IoT-blockchain applications have become increasingly popular around the world [2], such as Filament, Xage, and Atonomi. Filament [3] designs a microchip that integrates IoT products with blockchain technology to improve data security. Xage [4] provides a blockchain-based security platform to avoid data tampering in IoT devices. Atonomi [5] offers blockchain-based solutions such as immutable identity and reputation tracking to IoT devices. However, there are still some challenges when deploying IoT applications on blockchain systems due to limited storage, power, and computing capability of IoT devices. The majority of IoT-blockchain systems take Proof of Work (PoW) as their underlying consensus mechanism. However, since IoT devices have limited resources, it is hard for them to conduct expensive mining work [6]. The PBFT consensus protocol is considered as a suitable protocol for IoT systems. However, the PBFT algorithm can work well only in a small fixed-size network, in which participating nodes cannot freely join or leave. This may not satisfy the requirement of an IoT network that has numerous and dynamic nodes. Moreover, the PBFT protocol generates high traffic overhead due to frequent network communications ($O(n^2)$) among n nodes. Previous work utilizes location information in blockchains, such as [7]–[10]. Their work focuses on location accuracy, security, or privacy preserving. Seldom of them addresses consensus efficiency, network overhead, and blockchain scalability.

In this paper, we propose Geographic Practical Byzantine Fault Tolerance algorithm (G-PBFT), a novel consensus protocol for IoT-blockchain applications. The novelty is that we leverage the geographic information of IoT devices when running the PBFT algorithm to make the blockchain system immune to Sybil attacks. G-PBFT uses the genesis block to contain the geographic locations of some core IoT devices. They can be elected as endorsers to reach PBFT consensus. Consensus operations include transaction verification, block production, and block validation. If an endorser misses a block or causes a fork, it will be removed from the endorser list. A small number of endorsers have low consensus latency that makes the IoT-blockchain system scalable. Furthermore, we design an incentive mechanism with geographic features to encourage IoT devices to become endorsers.

The principle of the endorser election in G-PBFT is based

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

on the fact that most IoT-blockchain applications rely on fixed IoT devices, for example, a smart street lamp of a car monitoring system, or a payment machine in a parking lot. These IoT devices always have more computational power than other IoT devices such as mobile phones and sensors. Moreover, these fixed IoT devices may be owned by companies and probably will not be malicious nodes. Thus, G-PBFT is suitable in consortium or private blockchains.

The main contributions of this paper can be summarised as follows:

(1) We propose a novel location-based blockchain consensus protocol, G-PBFT, for IoT-blockchain applications. The protocol utilizes geographic information and timestamp from IoT devices to ensure the loyalty of endorsers and enhance the security of blockchain by avoiding Sybil attacks. The geographic information can be further used in the incentive mechanism to encourage IoT devices to become endorsers.

(2) The proposed G-PBFT is scalable for IoT applications because we form a small-size endorser committee to perform the consensus task. We select those powerful, loyal IoT devices as the endorsers to conduct the intensive consensus computation. Thus, the consensus can be obtained within a short interval even in a large-size IoT network. G-PBFT achieves high consensus efficiency and low network overhead.

(3) G-PBFT can efficiently handle endorser node arrival and departure with a new era switch mechanism in the paper, which is hard to solve in PBFT. With this mechanism, blockchain systems can quickly adapt to new network size and reach consensus.

(4) We develop a blockchain prototype with the G-PBFT protocol and conduct extensive experiments with around 200 endorsers (to support hundreds of IoT devices or even more). The results show that the G-PBFT protocol can reduce 97.8% consensus latency and 95.6% communication cost when compared to the traditional PBFT consensus protocol.

The rest of this paper is organized as follows. We first describe preliminary knowledge in Section II. Then, we detail the protocol design of G-PBFT in Section III. Theoretical analysis and experiment results can be found in Section IV and V, respectively. We summarize related work in Section VI. Finally, we conclude this paper in Section VII.

II. Preliminaries

Before elaborating on the design of G-PBFT, we introduce some preliminary knowledge in this section, including PBFT, IoT, and geographic information.

A. PBFT algorithm

Byzantine General Problem is a famous and intractable problem in distributed systems [11]. Another counterpart is Crash Failure Problem, which is simpler and more common. The Crash Failure Problem assumes that all nodes are honest. By contrast, the Byzantine General Problem implies a situation that there may be dishonest nodes in a distributed computing system. Specifically speaking, a dishonest node can send different or even contradictory messages to other nodes, aimed to prohibit a system from reaching a consensus or reach a false consensus. In the Byzantine General Problem, a system needs to reach information consensus among honest members and dishonest members. As a solution to solve the problem, Proof-of-work (PoW) is used in various blockchain systems, such as Bitcoin and Litecoin. However, PoW requires high computational power to maintain correctness of consensus.

Compared with the PoW algorithm, PBFT algorithm [12] is more lightweight and effective. It ensures correct consensus decision if the number of malicious nodes is less than 1/3 of total nodes. The workflow of PBFT algorithm can be divided into a succession of *views*. Three phases are involved in a *view* to commit a request:*pre-prepare*, *prepare*, and *commit*. In each view, there is only one node can be selected as the *primary*, and other nodes are called *backups*. In the *pre-prepare* phase, *primary* node broadcasts the *pre-prepare* message to each *backup* node. If a *backup* node accepts the *pre-prepare* message after verification process, it enters the *prepare* phase and multicasts the *prepare* message to all other nodes. The verification process mainly compares messages from different nodes, and it is considered valid if a node receives messages from more than 2/3 of total nodes and these messages contain a consistent data. Similarly, once a node (both the *primary* and *backups*) accepts the *prepare* messages, it enters the *commit* phase and broadcasts the *commit* message to all other nodes. Once the collected *commit* messages are considered valid, the node will give response to client. The client will make final decisions based on all the collected responses.

B. IoT network

IoT network can be considered as an extension of the existing Internet network, which aims to connect massive network-enabled devices. After accessing the IoT network, the devices can work automatically and intelligently without any manual intervention.

IoT devices include mobile phones, RFID tags, NFC devices, and various kinds of sensors. They can be seen in various areas for different services. For example, smart home appliances, smart cities, shared bicycles, electronic tickets (E-tickets), and mobile payments. However, there are still some problems limiting the development of IoT networks. As for a large number of IoT devices, it is hard to manage them in a centralized manner. Besides, many IoT devices are closely linked with user's privacy, such as healthy data collected by smart home appliances. The trends of IoT research is to improve network scalability and provide better privacy protection.

There are already some works trying to solve the above problems with the help of blockchain technology. Some typical IoT-blockchain systems are summarized in Table I, with comparisons of blockchain, consensus mechanisms, services, IoT devices, and company size.

The table shows that most of IoT-blockchain systems use Proof-of-Work approach. However, PoW is not compatible with IoT system when considering limitations of IoT devices. PoW requires high computational power and storage to

maintain credibility of blockchain. However, IoT devices are limited in processing capability, storage, and power. A suitable consensus mechanism for IoT-blockchain applications should achieve a balance between network overhead, computing overhead, and efficiency. The G-PBFT mechanism we proposed achieves high energy efficiency, high scalability, low network overhead, and low computing overhead.

C. Geographic information

With the increasing popularity of IoT devices, massive data is created every second. One of the most important and valuable data is geographic information, which records the real-world geographic information and timestamp of an IoT device [19]. The geographic information consists of two elements: longitude and latitude, which can be acquired via satellite-based radio navigation systems (e.g., GPS) or cell towers. As a result, a piece of geographic information usually has a format as $< longitude, latitude, timestamp >$.

Geographic information is useful in many scenarios. For example, by tracking the geographic information of our smartphones, the web mapping services (e.g., Google Maps) can guide us to destination. Besides, recommendations of nearby restaurants are usually made based on geographic information. Moreover, aggregated geographic information can be used in big data analysis. For instance, by analyzing collected geographic information of vehicles, the government can decide how many and how large parking lots should be built in a place.

III. Protocol Design

In this section, we elaborate on the design of G-PBFT protocol in detail. We firstly describe threat model and model assumptions. Then, we present an overview of the protocol, including introduction of different roles and terms. Finally, we interpret key components of the protocol step by step.

A. Threat Model

Less than 1/3 of total endorsers are faulty, either dishonest or frustrated. Cryptographic primitives took in our algorithm (e.g., public-key encryption) cannot be broken in a certain period. The adversaries can only create some invalid messages from itself, but cannot forge messages or tamper with the messages sent by others.

One of the crucial challenges of G-PBFT is faking of geographic information from adversaries. However, all IoT devices (including honest nodes and adversaries) are worked within a small physical area. Nodes can monitor and supervise each other, and check geographic information accordingly. By this theory, we can ensure that the geographic information reported by nodes is reliable.

B. Protocol Overview

The design of G-PBFT is based on the fact that the mainstream of IoT-blockchain applications rely on fixed IoT devices. For example, a wireless signal transmitter in a smart home system and a RFID receiver in a location tracking systems. These IoT devices always have more computational power than other IoT devices such as mobile phones, RFID tags, and smart systems in cars. Moreover, these fixed IoT devices may belong to management companies that will not become malicious nodes.

In the remaining parts of this section, we will first introduce some essential components and key terms in G-PBFT, and then give an overview of protocol workflow.

1) Node roles: Nodes are classified into two kinds of roles: endorser and client. The endorser participates in consensus. It maintains correctness of blockchain systems and proposed transactions. By contrast, the client only proposes new transactions to change the ledger status. All endorsers together make a consensus committee. Transactions will only be transmitted between the endorsers to reduce the communication overhead. In case of message crash, a client will send the transaction to multiple endorsers at the same time. The role of a node is not fixed. A client can apply to become an endorser. Once the node qualified by the committee, it can become an endorser. By contrast, if the location of a endorser changes or it is identified as faulty, it will be kicked out from the committee. The qualification policy of endorsers will be presented in Section III-D.

2) Transaction formats: IoT devices generate data and upload to blockchains for different application uses. Essential data will be treated as transactions such as temperature of temperature sensors, business data of mobile payments, signal strength of RFID tags. There are two kinds of transactions contained in our system, normal transactions and configuration transactions. Normal transactions are proposed by both clients and endorsers, which are used to change the ledger status on a chain for various application usages. For example, data of temperature sensors, business data of mobile payments, and signal strength of RFID tags can be sent to blockchains as normal transactions. By contrast, configuration transactions are used to modify chain configurations, such as adding new or deleting obsolete endorsers. Only current endorsers can propose this in the consensus committee. As already stated in Section II-C, both normal and configuration transactions carry the geographic information at the end of the transaction body.

3) Endorser election: Apart from the geographic information in transactions, our system requires IoT devices to upload their location and timestamp periodically. We use Crypto-Spatial Coordinates (CSC) to associate location information of IoT devices and its blockchain address. A CSC consists of location information (geohash) and a smart contract address. CSC is a hierarchical standard [20]. A shorter CSC address represents a larger area. A longer CSC addresses represent a more specific location. The resolution of CSC is about one square meter outside or inside a building. CSC helps IoT devices to make an immutable claim to its historical locations.

Before G-PBFT system performs endorser election, core nodes are considered as endorsers. They are responsible for validating and recording information from IoT devices. Endorsers store and maintain mapping of CSC and its timestamp

TABLE I: Comparison between IoT-Blockchain Applications

IoT-Blockchain	Blockchain	Consensus	Service	IoT devices	Company size
Atonomi [5]	Atonomi	Atonomi	IoT-blockchain solutions	Smart devices, Smart home	Leading provider of IoT data security
ElectriCChain [13]	SolarCoin	PoS	Process data of solar panel	Solar panel	Open source project
Filament [3]	Hardware-based Consortium Blockchain	PoW	Transaction service to embedded IoT	Blocklet USB Enclave, Blocklet Chip	40 milions market cap
JD.com [14]	BFT blockchain	BFT	Blockchain platform	IoT devices	1.7 trillions market cap
LeewayHertz [15]	Public blockchain	PoW	IoT-blockchain solutions	Robots, Audio devices	More than 10 years in operations
LO3 Energy [16]	Public blockchain solution	PoW	Solar energy marketplace	Grid Edge, Solar plane	1 million in revenue annually
Slock.it [17]	Ethereum	PoW	Commission shop	Electronic lock	1.5 millions in revenue annually
UniquID [18]	Litecoin	PoW	Integrated service to IoT and blockchain	Sensors, Actuators, Appliances	Open source project
Xage [4]	Fabric	PBFT	Security service	Broker, Enforcement Point	300 milions market cap

TABLE II: Election Table

	CSC	Timestamp	Geographic Timer
1	5AH71r9wTRp9eHsqR	5/8/2019 18:00:00	0
2	5AH71r9wTRp9eHsqR	5/8/2019 18:56:04	56:04
3	5AH71r9wTRp9eHsqR	6/8/2019 00:00:00	06:56:04
4	5AH71r9wTRp9eHsqR	6/8/2019 06:00:00	12:56:04
5	5AH71r9wTRp9eHsqR	6/8/2019 12:00:00	18:56:04

in an election table. An example of the election table is shown in Table II. Data uploaded from IoT devices to blockchains will add an entry to the election table. The second entry of the table is added by the uploaded transaction of IoT devices. Other entries of the table are added by periodic updates. Moreover, geographic timer in the election table will record how long an IoT device does not change its position. An IoT device stays at the same location (has the same CSC) for 72 hours will be elected as an endorser after an agreement of current endorsers.

After an IoT device is elected as an endorser, it starts to validate blocks, produce blocks, and pack transactions according to the G-PBFT consensus. If there are block missing and forking caused by an endorser, the endorser will be removed from the endorser list.

4) Views, phases and Eras: There are mainly three important terms related to time period, namely *views*, *phases*, and *eras*. Firstly, the former two are the terms similar to those in the traditional PBFT algorithm. More specifically, a running of PBFT algorithm can be divided into multiple *views*, and each *view* has one and only one *primary* node. Once the *primary* in a *view* crashes or behaves abnormally, another node will be elected as the new *primary*. Then, changing the *view* to the next *view*. In a *view*, each node can be in one of the three *phases*: *pre-prepare phase*, *prepare phase*, and *commit phase*, as introduced in Section II-A. Secondly, *era* is a new term introduced in G-PBFT. G-PBFT can be regarded as a splice of multiple successive PBFT, which schematic diagram is shown as Figure 1. Accordingly, an era denotes a period of PBFT with a fixed chain configuration. Once the chain configuration gets modified under consensus of endorsers, the chain will switch from one era to a new one.

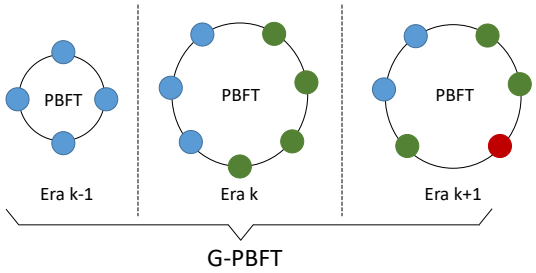


Fig. 1: Eras in G-PBFT.

5) Incentive mechanism: The incentive mechanism is an important part of blockchain to encourage blockchain nodes to ensure data correctness and integrity. It provides truthful services among participants without mutual trust. In our design, endorsers who generate a block or endorse a transaction can get their rewards. For an IoT device, the geographic timer in the election table is used for endorser election. For an endorser, the geographic timer is used for block generation. A longer time in the geographic timer will have a higher chance of generating a new block. Because an IoT device stays in a fixed location for a longer time represents a higher loyalty and honesty. Once an endorser successfully generated a block, its geographic timer will reset by the system. The reward incentive comes from transaction fee. An endorser generates a new block can get 70% of the transaction fee. Endorsers endorse others block can share 30% of the transaction fee.

If an endorser node missed a block or caused a fork, it will not be endorsed by other endorsers and get its rewards. This mechanism is secure against faulty endorsers who perform malicious actions.

6) Overall running: The running of G-PBFT algorithm consists of two stages: initiation stage and normal stage. At the initiation stage, core nodes will act as endorsers to initialize and launch the system, which presented in Section III-C. After initialization, the system enters the normal stage, where clients (i.e., IoT devices) generate data and send it to nearby endorsers, as shown in Figure 2. In each era, G-PBFT works similar to the PBFT algorithm, the ledger data change from

one state to another. In normal stage, blockchain nodes and IoT devices can enter and leave the blockchain network freely. A node can apply to become a new endorser after qualification examination. An endorser can be considered as invalid if it behaves abnormally. Qualification of endorsers will be introduced in detail in Section III-D.

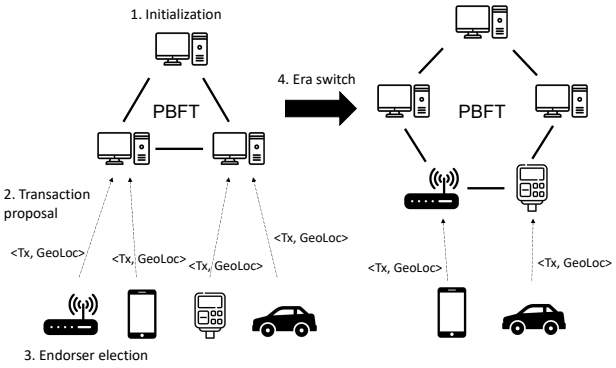


Fig. 2: Construction of G-PBFT.

C. System Initiation

At the beginning, there are core nodes assigned to consensus committee in blockchain. They are responsible for decision and consensus. At the stage of system initiation, multiple nodes are appointed as endorsers. Endorsers have to verify and store the identity information of each other, which is used to validate messages during the PBFT process. The information of the initiated endorsers is contained in the genesis block. It can be acquired by all nodes, including endorsers and clients. In the scenario of IoT-blockchain applications, endorsers are usually some devices with fixed locations, such as street lamps, traffic-control cameras, and base stations.

Besides, the genesis block contains extra admittance policies, such as blacklist, whitelist, minimum number, and maximum number of endorsers. Nodes in the blacklist will be forbidden to join the consensus committee. Nodes in the whitelist can be identified as endorsers directly without any qualifications. For the number of endorsers, if it is smaller than the minimum value, the system will stop accepting and committing new transactions. By contrast, if the number of endorsers exceeds the maximum value. The endorser election will be terminated until old endorsers leave. The maximum number and the minimum number are stored in the genesis block. Endorsers would not perform era switch if the number of endorsers reached the maximum value.

D. Geographic Authentication of Endorsers

To become a qualified endorser, a candidate has to pass qualification authentications. The process of qualification authentication consists of two aspects: PBFT-related authentication and geographic authentication. The former requires accordance between a node's public key and signature. The latter examines if the geographic information of the candidate is within the particular area and if a node has changed its location over a period of time, whose algorithm is shown as Algorithm 1.

Algorithm 1 mainly consists of two parts, lines 2-14 and lines 15-26, which can only be executed by endorsers. $\mathcal{V}$ and $\mathcal{C}$ represent endorsers and candidates, respectively. Lines 2-14 re-authenticate the qualification of members in the current consensus committee, where $G(v,t)$ is a chain-based function and returns the geographic information reported by a node during the past period t. If the number of geographic information reported is less than a threshold n, the endorser will be judged as invalid in next era. Besides, all geographic locations of a valid endorser must be the same. If not, the endorser will be judged as invalid too. Similarly, Lines 15-26 authenticate the qualification of new candidates. If all geographic locations of a candidate are the same over a certain period, it will become an endorser and be added to the consensus committee in next era. Algorithm 1 will be executed every T seconds, which avoids excessive calculation overheads.

Algorithm 1 Geographical location-related authentication of endorsers

```
1:  while IsEndorser() do
2:      for each v ∈ V do
3:          G ← G(v, t)
4:          if Len(G) < n then
5:              v[status] ← false
6:              continue
7:          end if
8:          for each g1, g2 ∈ G do
9:              if g1[lng] ≠ g2[lng] or g1[lat] ≠ g2[lat] then
10:                 v[status] ← false
11:                 break
12:             end if
13:         end for
14:     end for
15:     for each c ∈ C do
16:         G ← G(c, t)
17:         if Len(G) < n then
18:             continue
19:         end if
20:         for each g1, g2 ∈ G do
21:             if g1[lng] ≠ g2[lng] or g1[lat] ≠ g2[lat] then
22:                 break
23:             end if
24:         end for
25:         v[status] ← true
26:     end for
27:     sleep(T)
28: end while
```

E. Scalable by Era Switches

As stated in Section III-B4, G-PBFT can be considered as a combination of multiple successive eras, each of which is

an intact PBFT algorithm. New qualified candidates will start to work in next era, and disqualified endorsers will be kicked out after the finish of an era switch. The switch from one era to another one need to guarantee the system's security. G-PBFT algorithm works under the assumption that less than 1/3 endorsers are malicious, and each endorser makes decisions based on the majority mechanism independently. As a result, during the period of an era switch, the system will refuse to process or commit any transactions. The period used to switch an era is called 'switch period'.

Era switch will be made every T seconds in our system, which should be neither too small nor too large. A too-small T will lead to frequent era switches and many corresponding switch periods. Since the system cannot process transactions in the switching period, a too-small T will reduce the system performance of G-PBFT. By contrast, a too-large T will make the system unable to react to environmental change in time. As stated in Section III-C, a minimum value is set to ensure that there are enough endorsers in the system. The system cannot process transactions if there are not enough valid endorsers in the consensus committee. In this case, the system need to add new endorsers via an era switch. A too-large T will result in a long time of system pause, which reduces the system performance too.

By the mechanism of era switch, G-PBFT allows arrival and departure of IoT devices with a minimum impact on the system performance. Therefore, G-BFTT is able to achieve high network scalability over traditional consensus mechanisms.

IV. Theoretical Analysis

In this section, we analyze the G-PBFT algorithm theoretically, from three perspectives: security, performance, and overhead.

A. Security

In this section, we will analyze the security of G-PBFT. G-PBFT carry on the security advantages of original PBFT and improved to resist additional security risks.

1) Sybil attack: Traditional Blockchain suffers from Sybil attack and Sybil nodes created by malicious users. Since nodes can enter the system without any permission mechanism, a malicious user may spawn massive Sybil nodes. This aim at controlling PBFT consensus process. As long as there are enough Sybil nodes enter the consensus committee (e.g., more than 1/3 of the endorsers), malicious users can do evil on ledger data.

To address this problem, the system requires nodes to report its geographic information periodically. On the one hand, different nodes cannot report the same geographic information at the same time. This limits the maximum number of Sybil nodes in an IoT-blockchain system. On the other hand, since all IoT devices of an IoT-blockchain application are located in a small physical area, other nodes can easily identify the fake geographic information reported by a malicious user. For example, if there is no device in a specific position and geographic information reporting, it can be recognized as fake.

2) Era switch: Another security problem is related to the era switch. In G-PBFT, an endorser makes decisions based on ratio of the number of valid messages to the number of total endorsers. However, there may be change of endorsers number due to arrival or departure of IoT deivces in an IoT system. In this regard, if the number of endorsers is N in the last era, an endorser will switch from $preparephase$ to $commitphase$, once it receives more than $N/2$ $prepare$ messages. Assume that there are $2N$ incoming endorsers in the new era, and an endorser A in the last era remains in a new era. A will continue to switch from one phase to another to ensure the number of valid endorsers until it received the latest committee information. As a result, G-PBFT asks each endorser to halt the old consensus before era switch. Also, it relaunches the new consensus after the finish of the era switch, thus making the number of total endorsers in each era constant.

B. Performance

The main innovation of G-PBFT is to elect out a consensus committee to run PBFT, which can adapt to the change of the IoT network. Since the scale of the consensus committee in G-PBFT is much smaller than the entire IoT network, the consensus performance is expected to improved largely. In this section, we try to make a quantitative analysis of the performance improvement brought by G-PBFT.

Let n and c represent the number of total IoT nodes and endorsers in G-PBFT, respectively. Let s represent the processing power of a node, which means a node can receive and process s messages per second. With the original PBFT as the consensus algorithm, each node has to receive at least $(2*n)/3$ messages to change from one phase (e.g., *pre-prepare phase*) to another (e.g., *prepare phase*). As a result, it takes at least $(2 * n)/(3 * s)$ seconds to finish a phase switch. In the same way, it takes at least $(2 * n)/(3 * s)$ seconds to switch from *prepare phase* to *commit phase*. To sum up, a complete consensus process in PBFT will take at least $O(n/s)$ seconds. Similarly, a complete consensus process in G-PBFT takes at least $O(c/s)$ seconds. Therefore, the time to reach a consensus can be reduced to c/n. In other words, the performance can be improved by n/c. In addition, the larger ratio of the number of total IoT nodes to endorsers is, the greater the performance improvement will be.

C. Overhead

One of the most important reasons for PBFT's poor scalability is its high communication overhead. To be specific, a node has to broadcast a message to all other nodes and receive valid messages from at least $2/3$ nodes. As a result, the communication overhead of PBFT is about $O(n^2)$, where n is the number of total IoT nodes. As evaluated in [21], the PBFT algorithm can only scale to 16 nodes in the hyperledger system.

G-PBFT algorithm reduces the scale of the consensus committee, it is expected to reduce the communication overhead. Since a node in G-PBFT only needs to send messages to other endorsers rather than all nodes, the communication overhead

of PBFT is about $O(c^2)$, where c represents the number of endorsers. As a result, G-PBFT can reduce the communication overhead by $(c^2)/(n^2)$. The larger ratio of the number of total IoT nodes to endorsers is, the larger the reduction of communication overhead will be.

V. Evaluation

To evaluate our design, we develop a prototype of a blockchain system with G-PBFT as consensus protocol. To make comparisons with PBFT, we conduct experiments of original PBFT and G-PBFT. In this section, we present experiment results of system performance and network overhead.

A. Experiment Setup

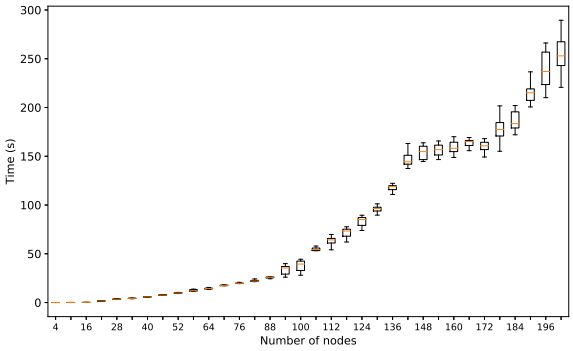
We conduct experiments on our server machines, each of them contains a two-core Intel Core i7 2.2 GHz CPU with 16GB DRAM and 256GB SSD, with Ubuntu 16.04 as operating system. We simulate server machines as IoT devices in our experiments. The initial consensus committee consists of 4 IoT devices, and the minimal and maximal values stated in Section III-C is set as 4 and 40 separately. There are numerous IoT nodes in an IoT system. However, the number of nodes participating in the endorser committee could be small. In our experiment, we use a reasonable amount of 202 nodes as the number of endorsers in the committee, which should facilitate the running of a large IoT network.

B. Performance

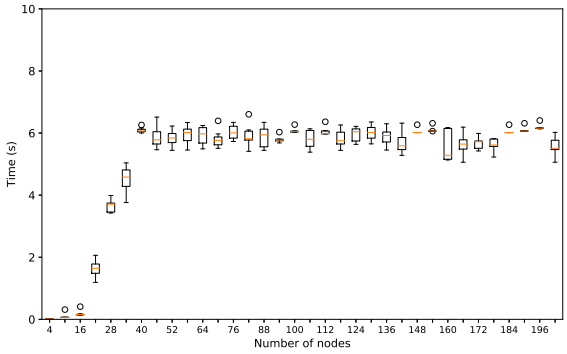
Instead of measuring the Transactions Per Second (TPS) of the blockchain system, we evaluate the performance in terms of consensus latency to commit a transaction. In other words, we measure the latency from the time when a transaction is sent to an endorser to the time when the transaction is written to the ledger after consensus. Each node is set to propose new transactions at a constant frequency. The number of nodes participating in the blockchain is increased from 4 to 202.

As a comparative experiment, we also evaluate consensus latency of PBFT. For a certain number of nodes, we run ten experiments to eliminate possible errors. The experimental results are shown in Figure 3a. For the ten experiments in each group, we draw a boxplot to display the distribution of data. Upper and lower lines represent the maximum and minimum values, respectively. The line in rectangles denote the median value, while the upper and lower side indicates the third and first quartiles. It is easy to find that as the number of nodes increases, the consensus latency increases at an exponential speed accordingly. Besides, variances increase in general, which indicates that there is great uncertainty about consensus latency.

By contrast, G-PBFT shows a better performance in terms of consensus latency, whose experimental results are depicted in Figure 3b. When the number of nodes is smaller than the maximal value of endorsers (i.e., 40), all eligible nodes can join the consensus committee. As a result, the consensus latency increases just like that in the PBFT consensus. However, once the number of nodes reaches the maximal value,



(a) PBFT consensus latency



(b) G-PBFT consensus latency

Fig. 3: Consensus latency with different number of nodes

no more endorsers will be added into the committee, and the consensus latency will not increase anymore. Futhermore, the variance is much smaller in G-PBFT consensus, which enables a transaction to be committed within a steady period. It should be noted that there may be a circle for a certain number of nodes in Figure 3b, which is an outlier in a group of data. The reason for it is the time taken to finish an era switch is about 0.25 second.

To compare two consensus algorithms in more detail, we further increase the number of nodes and calculate the averages of consensus latency for different numbers of nodes.

As shown in Figure 4, when the number of nodes reaches 202, the consensus latency of G-PBFT can still keep at a stable small value. On the other hand, when the number of nodes reaches 202, the consensus time of PBFT is over 250 seconds. Table III compares the average consensus latency between PBFT and G-PBFT. When the number of nodes is 202, G-PBFT reduces the consensus latency to 2.24%.

To sum up, these experimental results demonstrate a better performance of G-PBFT consensus mechanism.

C. Communication costs

As stated in Section IV-C, G-PBFT is expected to reduce communication costs largely, especially when the number of IoT devices is much larger. To evaluate the reduction of com-

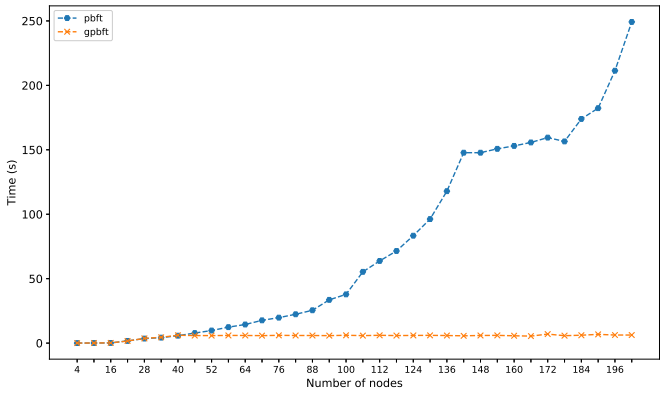


Fig. 4: Comparison of consensus latency between PBFT and G-PBFT.

TABLE III: Experimental results when number of nodes is 202

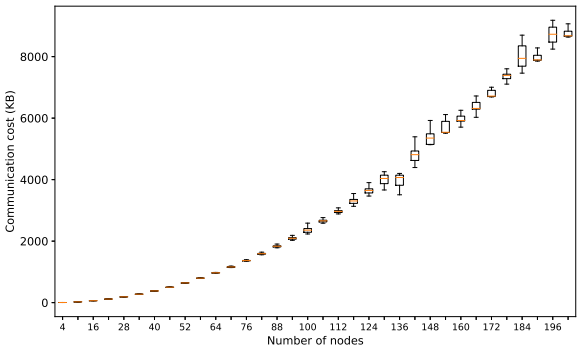
Consensus	Average latency (s)	Average costs (KB)
PBFT	251.47	8571.32
G-PBFT	5.64	380.29

munication costs by G-PBFT, we conduct multiple groups of experiments with the number of nodes varying in this section. Different from the experiment setting in Section V-B, which asks each node to propose transactions constantly, we only propose one transaction in each experiment here. As a result, communication cost is evaluated for a single transaction.

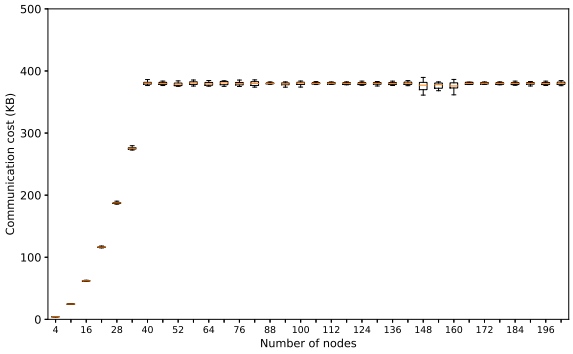
Figure 5a and Figure 5b depict the experimental results of PBFT and G-PBFT respectively. As can be seen in Figure 5a, communication cost in PBFT algorithm keeps increasing when the scale of IoT network is enlarged. Besides, the larger is the number of nodes, the quicker is the increase in communication cost. By contrast, the communication costs reach an upper boundary of about 400KB, even if the number of nodes is over 100.

Figure 6 compares the communication costs of PBFT and G-PBFT more clearly. Similar to Section V-B, PBFT network cannot work at all when the number of nodes is larger than 202. As a result, the line representing PBFT breaks after the x-axis is over 202. Also, Table III compares the average communication costs between PBFT and G-PBFT, when the number of nodes is 202. G-PBFT can reduce costs to 4.43%.

Both Figure 5 and 6 show a smaller communication costs brought by G-PBFT, as stated in Section IV-C.



(a) PBFT communication costs



(b) G-PBFT communication costs

Fig. 5: Communication costs with different number of nodes

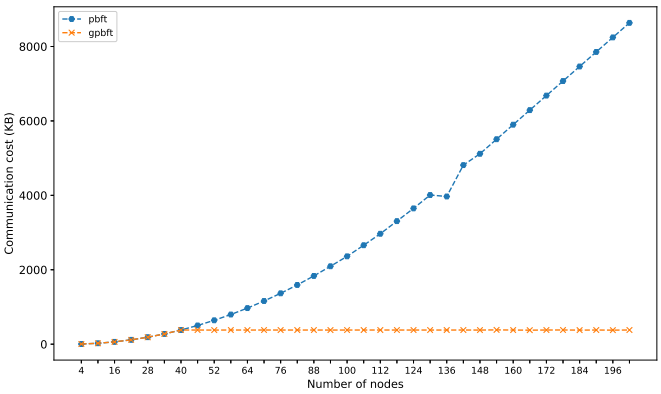


Fig. 6: Comparison of communication costs between PBFT and G-PBFT.

VI. Related Work

In this section, we summarize the related works on consensus mechanisms and IoT-blockchain applications.

A. Consensus Mechanisms

In blockchains, a consensus mechanism is essential to ensure data correctness among participants without mutual trust.

Byzantine Fault Tolerance (BFT) [22] is a mechanism to reach consensus in a system with certain faulty participants. It solved the *Byzantine General Problem* [11]. BFT is a replica-based approach that utilizes communications between replicas to reach consensus.

Practical Byzantine Fault Tolerance (PBFT) [12] is practical solution of the BFT mechanism. It enhances system performance, reduces the complexity of *Classic Byzantine General Problem* to polynomial level, and can work properly in a distributed system. PBFT requires at least $3f + 1$ participants

TABLE IV: Comparison between consensus

Consensus	Blockchain type	Speed	Scalability	Network Overhead	Computing Overhead	Adversary Tolerance	Example of use
BFT	Permissioned	High	Low	High	Low	<33.3% Replicas	Tendermint
PBFT	Permissioned	High	Low	High	Low	<33.3% Faulty Replicas	Hyperledger
dBFT	Permissioned	Low	High	High	Low	<33.3% Faulty Replicas	NEO
PoW	Permissionless	Low	Low	High	High	<25% Computing Power	Bitcoin
PoS	Permissionless	Low	Low	High	Low	<50% Stake	Peercoin
DPoS	Permissionless	High	Low	Low	Low	<50% Validators	BitShares
PoA	Permissionless	Low	High	Low	Low	<50% of Online Stake	Decred
PoSpace	Permissionless	Low	Low	High	Low	<50% Space	SpaceMint
PoI	Permissionless	Low	Low	High	Low	<50% Stake	NEM
PoB	Permissionless	Low	Low	High	Low	<50% Coins	XCP
G-PBFT	Permissionless	High	High	Low	Low	<33.3% Endorsers	

in order to tolerate f faulty nodes.

Delegated Byzantine Fault Tolerance (dBFT) [23] is proposed by a cryptocurrency NEO. It determines the consensus committee by real-time blockchain voting. dBFT is based on PBFT and has similar features with it. However, dBFT increases the scalability of PBFT by delegated nodes.

However, the BFT and PBFT consensus protocols still have their limitations. For example, they have poor scalability that participants cannot dynamically join or leave blockchain networks. Moreover, the average latency of dBFT to produce a block is 15 seconds, which is not suitable to use in IoT systems [24].

Proof-of-work (PoW) was proposed by Nakamoto *et al.* in 2008 [25]. It avoids denial of service attacks (DoS) and malicious actions by requiring participants to conduct mining work. Mining implies to solve a computationally complicated mathematical problem which requires huge computational power. Through this, PoW maintains the correctness of blockchain. However, PoW is vulnerable to various mining attacks [26].

Proof-of-Stake (PoS) [27] relies on an assumption that participants holding more currency are more reliable to ensure system's validity, and is less likely to conduct malicious acts. In PoS [28], participants holding more currency over a long time have a higher priority of being selected by the community to generate new blocks. In DPoS [29], a supernode is elected to generate new blocks. DPoS increases system performance by reducing the number of nodes in consensus. However, it is not fully decentralized because the decentralization process only occurs in supernode election.

Proof-of-Authority (PoA) [30] depends on authority instead of the amount of asset or computational power. In PoA, only nodes with authority are permitted to generate new blocks.

Proof-of-Space (PoSpace) [31], also known as Proof-of-Capability (PoC), makes use of disk storage to provide proof that a user has paid the price to compete to produce a new block. Concretely speaking, a user stores a piece of data according to its public key, to be a prover. A verifier will send multiple challenges to the prover afterward, to verify if the latter stores data honestly. As proof of the challenge, the prover will return a Merkle proof to the verifier. As the disk space is used meaningfully, PoSpace is considered as being more economical and environment-friendly than the PoW algorithm.

Proof-of-Importance (PoI) [32] tries to do some evaluations and give a corresponding mark for each node. It makes use of the mark to elect an eligible node to add a new block. The evaluation is conducted from various aspects, including an account' amount, number of relevant transactions, number of transaction partners. Different from PoW and PoSpace, PoI does not need a user to consume any resources, even if it is offline.

Proof-of-Burn (PoB) [33] is a consensus, which is more closely connected to cryptocurrency economic. A node has to burn some coins to compete as a block producer. By sending coins to an unspendable address and providing a corresponding proof, a node can prove to others that it has burned a certain amount of coins.

We summarize the characteristics of some widely-deployed consensus mechanisms above in table IV. In the table, we compare the type of blockchain, transaction speed, scalability, network overhead, computing overhead, tolerated power of adversary and existing applications. From the table, we can see that the G-PBFT mechanism has advantages of high transaction speed, high scalability, low network overhead, and low computing overhead over other consensus mechanisms.

B. IoT-blockchain Application

IoT-blockchain applications are still at an early stage. With the increasing popularity of IoT applications, there are more and more attempts to combine blockchain with IoT.

LO3 Energy [16] introduces a Peer-to-peer market for buying or selling solar energy. The IoT devices in this system include electrical grid and solar panels. Sellers itemize their extra energy yielded from solar panels and puts them on blockchains. Buyer can purchase green energy by the corresponding distributed applications (DApps). "Slock.it" [17] is an existing IoT-blockchain application. The IoT device of this system is smart locks that can be unlocked by DApps. Sellers can set a specific price on an electronic lock that associate with their properties. Buyer can browse the commodity and pay the price in cryptocurrencies to unlock the lock. Sagirlar *et al.* propose an IoT-blockchain platform "Hybrid-IoT" [34]. Hybrid-IoT implements consensus based on PoW and BFT algorithms. Bahga *et al.* introduce a platform for industrial IoT (BPIIoT) [35]. Users can develop distributed applications on a single-board computer (SBC) to control and manage

IoT devices through a blockchain network. JD Blockchain Open Platform [14] from JD e-business company focuses on providing integrated IoT and blockchain solutions. The platform provides node management, blockchain gateway, and BFT consensus service. Alphand *et al.* [36] introduce "IoTChain" which integrates OSCAR architecture and ACE authorization framework. In IoTChain, authorized token of every registered participant identifies the particular privilege of resources. Participants are required to send a transaction with the requested data to the smart contract address to access a specific object. Then the smart contract will generate the corresponding authorized token. This architecture uses blockchain instead of centralized ACE authorization server.

VII. Conclusion

With the increasing popularity of IoT-blockchain applications, the performance and scalability of IoT-blockchain systems become more and more critical. Aiming at solving the poor scalability and high overhead in existing IoT-blockchain applications, we propose G-PBFT, a location-based and scalable consensus protocol. The proposed G-PBFT achieves high consensus efficiency, low network overhead, and high scalability by location-based endorser election and era switch mechanism. We select those geographically distributed, powerful, and authenticated IoT devices in the endorser committee to take the consensus role. The proposed protocol can protect IoT-blockchain systems from Sybil attacks by using authenticated geographic location to filter out Sybil nodes. Extensive experiments are conducted to indicate superior performance of G-PBFT over traditional PBFT consensus mechanism.

VIII. Acknowledgements

This work was supported in part by the HK RGC GRF PolyU 152124/19E, HK PolyU H-ZG6Y, and HK ITF ITS/081/18.

References

[1] R. Jackson, "Why IoT needs the blockchain, and blockchain needs IoT," Available at https://hackernoon.com/why-iot-needs-the-blockchain-and-blockchain-needs-iot (2019/09/30).

[2] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of IoT applications in blockchain systems: Architecture, consensus and traffic modeling," *ACM Computing Surveys*, 2020.

[3] Filament, "Filament's industrial internet of things blockchain solution wins 2018 IoT innovator award," Available at https://globenewswire.com/news-release/Filament-2018-Award.html (2019/09/30).

[4] Xage, "Xage security," Available at https://xage.com/ (2019/09/30).

[5] Atonomi, "Atonomi - bringing trust and security to IoT," Available at https://atonomi.io/ (2019/09/29).

[6] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, "Resource allocation and consensus on edge blockchain in pervasive edge computing environments," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1476–1486.

[7] G. Brambilla, M. Amoretti, and F. Zanichelli, "Using blockchain for peer-to-peer proof-of-location," *arXiv*, 2016.

[8] F. Corp, "Foamspace Corp. FOAM whitepaper," Available at https://foam.space/publicAssets/FOAM_Whitepaper.pdf (2019/09/30).

[9] S. Migliorini, "Enhancing blockchain smart-contracts with proof-of-location," in *10th International Conference on Geographic Information Science*, 2018.

[10] L. Bornholdt, J. Reher, and V. Skwarek, "Proof-of-location: A method for securing sensor-data-communication in a Byzantine fault tolerant way," in *Mobile Communication-Technologies and Applications; 24. ITG-Symposium*, 2019, pp. 1–6.

[11] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[12] M. Castro, B. Liskov *et al.*, "Practical Byzantine fault tolerance," in *OSDI*, 1999, vol. 99, pp. 173–186.

[13] ElectriCChain, "Electricchain the solar energy blockchain project for climate change and beyond," Available at https://www.electricchain.org/ (2019/09/30).

[14] "JD enterprise blockchain service," Available at http://blockchain.jd.com/ (2019/09/30).

[15] LeewayHertz, "Blockchain development for startups and enterprises," Available at https://www.leewayhertz.com/ (2019/09/30).

[16] LO3, "LO3 energy the future of energy," Available at https://lo3energy.com/ (2019/09/30).

[17] SlockIt, "SLOCK.IT," Available at https://slock.it/ (2019/09/30).

[18] UniquID, "UniquID incorporation blockchain identity access management," Available at https://uniquid.com/ (2019/09/30).

[19] G. Djuknic and R. Richton, "Geolocation and assisted GPS," *Computer*, vol. 34, no. 2, pp. 123–125, 2001.

[20] M. N. K. Boulos, J. T. Wilson, and K. A. Clauson, "Geospatial blockchain: promises, challenges, and scenarios in health and healthcare," *International Journal of Health Geographics*, 2018.

[21] T. Dinh, J. Wang, G. Chen, R. Liu, B. Ooi, and K. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.

[22] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg, "Byzantine fault tolerance, from theory to reality," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2003, pp. 235–248.

[23] NeoDocsBuilder, "NEO consensus mechanism," Available at https://docs.neo.org/docs/en-us/basic/technology/dbft.html (2019/09/30).

[24] M. Salimitari and M. Chatterjee, "An overview of blockchain and consensus protocols for IoT networks," *arXiv*, 2018.

[25] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[26] S. Gao, Z. Li, Z. Peng, and B. Xiao, "Power adjusting and bribery racing: Novel mining attacks in the bitcoin system," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 833–850.

[27] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.

[28] H. Azima, "NXT white paper," *Huntingdon PE18 6ED Harris, N*, 2018.

[29] D. Larimer, "Delegated proof-of-stake (DPoS)," *Bitshare whitepaper*, 2014.

[30] A. Naumoff, "Why blockchain needs 'proof of authority' instead of 'proof of stake'," Available at https://cointelegraph.com/news/why-blockchain-needs-proof-of-authority-instead-of-proof-of-stake (2019/09/30).

[31] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Advances in Cryptology – CRYPTO 2015*. Springer, 2015, pp. 585–605.

[32] NEM, "NEM whitepaper," Available at https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf (2019/09/30).

[33] J. Frankenfield, "Proof of burn," Available at https://www.investopedia.com/terms/p/proof-burn-cryptocurrency (2019/09/30).

[34] G. Sagirlar, B. Carminati, E. Ferrari, J. D. Sheehan, and E. Ragnoli, "Hybrid-IoT: hybrid blockchain architecture for internet of things-pow sub-blockchains," in *2018 IEEE International Conference on Internet of Things (iThings)*, 2018, pp. 1007–1016.

[35] A. Bahga and V. K. Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 9, no. 10, pp. 533–546, 2016.

[36] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, and F. Zanichelli, "IoTChain: A blockchain security architecture for the internet of things," in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. IEEE, 2018, pp. 1–6.