On the Consensus Mechanisms of Blockchain/DLT for Internet of Things

Qingqiang He[1], Nan Guan[1], Mingsong Lv[2] and Wang Yi[2,3]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
[2]Smart Systems Lab, Northeastern University, China
[3]Department of Information Technology, Uppsala University, Sweden

***Abstract*—Internet of Things (IoT) has been experiencing exponential growth in recent years, but still faces many serious challenges. The distributed ledger technology (DLT), e.g., Blockchain, not only appears to be promising to address these technical challenges, but also brings tremendous opportunities for new application and business models. However, the convergence of IoT and DLT is yet a goal far beyond our reach today. Among many problems that have not been sufficiently understood, a fundamental one is how to design appropriate consensus mechanisms for DLT applied to IoT, which is the theme of this paper. We first discuss the potential benefits of applying DLT to IoT, and identify major challenges posed to DLT by IoT. Then we make a survey of existing DLT consensus mechanisms, to summarize major principles and discuss their pros and cons when applied in IoT.**

***Index Terms*—Blockchain, Distributed Ledger Technology (DLT), Internet of Things, Consensus Mechanism**

I. Introduction

Internet of Things (IoT) is a worldwide network of interconnected objects and human beings, which through unique addressing schemes are able to interact with each other and cooperate with their neighbors to reach common goals [1]. IoT has a wide application prospect in many domains, such as agriculture, manufacturing, consumer technology, and almost all mechanically intensive industries. However, at its current stage, IoT still faces serious challenges in many aspects, such as security, privacy, scalability, and maintainability.

Since the start of Bitcoin [2], especially since its fast growth in 2013 and 2014, Blockchain has emerged as a very attractive technology that promises tremendous potential for creating new application and business models. Blockchain [2], a type of distributed ledger technology (DLT), records transactions and is maintained by many nodes without a central authority through a distributed cryptographic protocol. All nodes validate the information to be appended to the Blockchain, and a *consensus mechanisms* ensures that the nodes agree on a unique order in which transactions are appended. Many people consider Blockchain to be a technological breakthrough, because it is for the first time in history that humans developed a system to reliably coordinate actions among many parties without having any central authority [3]. Although Bitcoin is the most successful application of Blockchain (and probably the only one when this article is written), Blockchain has potentials in a wide range of application areas beyond cryptocurrencies, such as finance [4], healthcare [5], reputation system [6].

978-1-5386-4155-2/18/$31.00 ©2018 IEEE

The Blockchain/DLT technology has many attractive features, such as decentralization, persistency, anonymity and auditability [7]. These features make Blockchain/DLT a promising solution to address the challenges in IoT. Both academia and industry [3], [8]–[12] have started to look at the application of Blockchain/DLT to IoT, not only to solve the problems faced by IoT, but also for potentials in new application paradigms and business models. However, the convergence of IoT and DLT is yet a goal far beyond our reach today. There are a lot of fundamental problems that have not been sufficiently understood. A fundamental one is how to appropriate consensus mechanisms Blockchain/DLT applied in IoT. The consensus mechanism is a core component in the design of Blockchain/DLT, which largely decides the performance, scalability, security and many other aspects of the system. In this paper, we will discuss the problems to be considered in the design of consensus mechanism of Blockchain/DLT for IoT, and review the existing consensus mechanisms and examine their applicability for IoT.

The rest of this paper is organized as follows. In Section II, we present the benefits that Blockchain/DLT technology can bring to IoT, and the important issues to consider when applying Blockchain/DLT in an IoT environment are discussed in Section III. In Section IV and V, we review existing DLT consensus mechanisms, summarize major principles and discuss their pros and cons when applied in IoT. Section VI concludes this paper.

II. What Blockchain/DLT can Do for IoT

A. Efficiency

The decentralized architecture of Blockchain/DLT can improve system efficiency in several aspects by using resources of all participating nodes [9].

1) Deployment and Maintenance Efficiency: The centralized cloud, network infrastructures, and large server clusters usually incur high deployment and maintenance cost, which render existing IoT solutions expensive. Blockchain/DLT offers an elegant solution to the peer-to-peer communication platform problem thanks to their distributed nature. Instead of an expensive, centralized data center, a data storage network utilizing Blockchain technology is duplicated across hundreds (even thousands or millions) of computers and devices. This

huge amount of redundancies make data to be always close at hand, which reduces both the transmission delay and the management overhead.

An example is utilizing Blockchain/DLT to reduce maintenance cost for firmware update in IoT [8]. With Blockchain/DLT, the manufacturer can deploy a smart contract [13] to store the hash of the latest firmware update on the network. All devices in the network can then query the contract, discover the update, and request it via a distributed peer-to-peer filesystem such as IPFS [14]. The advantage of this approach is that devices that join the network long after the manufacturer has stopped participating, can still receive the authentic file. The whole process works automatically, without any user interaction.

2) The Cost of Intermediaries: The use of Blockchain/DLT technology can reduce the overhead cost of intermediaries when several parties trade assets and services directly with each other. Blockchain and smart contract give rise to the concept of decentralized autonomous organization (DAO) [15], which can manage the exchange and supply of paid data and services, using cryptocurrency as the medium of trading. This removes the participation of third parties. Participants and even devices can buy data from even a single sensor directly. The removing of intermediaries brings opportunities to service sharing in IoT network, and may unlock new paradigms of machine-to-machine economy [16].

B. Reliability

Reliability is a critical aspect of IoT, especially for those serving for safe-critical applications. The beauty of IoT is that it automates mundane work so that our interconnected lives are simpler and more efficient. If the reliability of IoT is compromised and the glitches of IoT system continue interrupting its users, the value of IoT will be certainly affected. This decentralized nature of Blockchain/DLT can increase reliability by removing the single-point of failure, which is an inherent problem of centralized systems.

Besides, Blockchain technology can be used to increases the reliability of data integrity service in an IoT network through eliminating the trust requirements on third party auditors, while ensuring data integrity for cloud-based IoT applications can be very challenging because of the inherently dynamic nature of IoT data. In [17], a Blockchain-based framework for increasing the reliability of data integrity service is proposed by replacing integrity management service from the centralized node with a fully decentralized Blockchain based service.

C. Security

Security in IoT is a challenging issue due to low resource capabilities of the vast majority of devices, the extremely large scale of the network, high device heterogeneity and lack of standardization [18]. Blockchain/DLT can potentially provide trust, auditability, transparency, which enhances the security of IoT.

1) Trust among Devices: Authentication, connection, and transaction are three typical areas where IoT security flaws can arise. In an IoT environment, interactions happen between known or unknown devices, and devices improperly verifying, improperly connecting, or improperly spending with other devices poses major threats to IoT security. Consider the following scenario concerning autonomous machine repair, which is a big goal for the autonomous industry: when signs of deterioration or mechanical failures are detected, certain nodes of the network must respond to these events by ordering new parts or calling for a repair service. There are major attack vectors against such communications and connections in an IoT network. This problem can be solved by the trustless, consensus mechanisms of the Blockchain/DLT.

2) Auditability: One of the requirements of IoT is to track and verify the data flow and operations of network components. Performance analysis, network security, and legal compliance can benefit from such auditability. The immutability of records makes Blockchain an ideal option for creating reliable networks histories. In an IoT environment with Blockchain deployed, sensor data can be tracked, duplication of malicious data can be prevented. What's more, IoT devices can be uniquely identified in a distributed leger, and a history of connected devices thus provided can be used for troubleshooting purposes. In [19], a Blockchain-based design for auditable storage and sharing of IoT data was proposed, which makes use of the auditability of Blockchain and designs an auditable access control to IoT data.

3) Identity and Access Management: The difficulty of ensuring that physical assets, individuals, resource use and other relevant events are stored and accessed securely and reliably is a key challenge arising is some applications. In this respect, Blockchain can be used to handle the difficulty relatively easily. For example, device firmware can be stored in a private Blockchain, through which, a permanent and auditable database of device configuration can be established. In this setting, when a device attempts to connect to other devices or services, such database can be used to verify its firmware and configuration are not tampered with.

An example with respect to identity and access management in IoT is related to the distributed denial of service (DDOS) attack in which the attacker uses several infected IoT devices to overwhelm a particular target node [20]. Blockchain-based identity and access management systems can provide defense against such attacks. Because it is not possible to tamper with approved Blockchains, it is not possible for devices to connect to a network by injecting fake signatures into the database [21].

D. Privacy

Besides security, privacy is another serious concern for the current IoT systems. Conventional privacy preserving methods rely on revealing noisy or summarized data to the data requester [22]. In contrast, several IoT applications require users to reveal precise data to the service providers to receive personalized services [23]. Blockchain/DLT technology brings

a promising approach to the privacy issue in an IoT environment because of its decentralization and anonymity natures. With decentralization, personal data are stored in a distributed public ledger, which is not under the control of any central authority. Since there are no third parties who collect and control massive amounts of personal data, users shall not be afraid of the compromise of privacy, which happens in the centralized model [24]. With respect to Anonymity, as stated in [9], the inherent anonymity provided by Blockchain is suited for some IoT use cases, such as healthcare [5], where the identities of the users must be kept private. With the identities of users being hidden, privacy can be preserved.

E. Service Sharing

The use of Blockchain/DLT and smart contract also opens up new opportunities for IoT. A Blockchain network with cryptocurrency exchanged provides a convenient billing layer and paves the way for a marketplace of services between devices [8]. For example, EtherAPIs [25], where the callers pay fees using micropayment [26] before requesting these APIs, makes API calls a valuable commodity. Filecoin [27], as an open-source, public digital payment system, functions as Blockchain-based digital storage which allows devices to rent their disk space. TransActive Grid [28] is trying to realize the concept of peer-to-peer energy transaction and control, which may enable nodes to buy and sell energy automatically, using smart contracts and the Blockchain. Cryptocurrency and Blockchain make every node in an IoT network possessing a bank account possible, through which nodes can expose its services and resources to the network and get paid via micropayment.

III. Challenges of the Deployment of Blockchain/DLT in IoT

Since the advent of Bitcoin, Blockchain technology has received tremendous attention, investment and development activity in the past few years. But Blockchain/DLT is still in its nascent stage, and there are a range of challenges for the deployment of Blockchain/DLT in IoT. We highlight the major ones in this section.

A. Scalability

As IoT networks are expected to contain a large number of nodes, the ability to scale to satisfy the service and performance requirements among a dynamic network of devices is a critical challenge facing Blockchain/DLT being deployed in IoT . The issue of Blockchain scalability is related to the following four aspects.

1) Transaction Throughput: Due to the original restriction of block size and the time interval used to generate a new block, the Bitcoin Blockchain can only process nearly 7 tps (transactions per second), and at its max, Ethereum can handle 25 tps. By contrast, the transaction throughputs of VISA and Twitter are 2,000 tps and 5,000 tps, respectively. In IoT networks, millions of connected devices can communicate and transact simultaneously, which necessitates a high transaction throughput.

2) Transaction Latency: In Bitcoin protocol, the *block time* (average time requiring to mine a block) is 10 minutes. This 10 minutes block time is necessary to secure the whole network, primarily to prevent the *double spending attack*. Double spending is the result of successful spending of money more than once [29]. However, in order to make sure that a transaction is confirmed, 6 blocks being mined after the transaction being included in a block is recommended [30]. For some important transactions which require a high confirmation confidence (i.e. the possibility of a transaction being confirmed), even more blocks have to be mined. This makes transaction latency even longer in Bitcoin Blockchain. Confirming a transaction while ensuring security of the network should happen in seconds. The transaction latency of VISA is only a few seconds, which is a huge advantage over Blockchain [31]. What's more, in Bitcoin network, since miners prefer transactions with a high transaction fee, micropayment, which may play an important role in service sharing of IoT [32], may be delayed for a longer period.

3) Network Bandwidth: In Blockchain network, every transaction and every confirmed block need to be broadcast across the whole network, which may occupy a large amount of network bandwidth, which is undesirable for certain bandwidth limited IoT devices.

4) Storage: Blockchain/DLT is designed to be a distributed database, and for validating the transaction and ensuring data integrity, the whole database has to be stored on each node in the network. However, devices in the IoT network are typically not equipped with large storage space. With the number of transactions increasing day by day, The size of Blockchain will increase with new transactions. Currently, Bitcoin Blockchain has exceeded 100 GB storage [33], which is certainly beyond the capability of most of IoT devices.

B. Resource Utilization

As mentioned in Section II-A, compared to centralized architectures of IoT, a decentralized architecture like Blockchain can reduce the overall cost of the system. However, Blockchain/DLT technology introduces a new type of resource wasting, which poses great challenges for Blockchain being deployed in IoT. In a centralized architecture, consensus is ensured by a trusted authority. While in a decentralized environment, nodes of the network need to reach consensus by voting, which is a resource-intensive process. IoT devices are characterized by relatively low computing capabilities and low power consumption, as well as sporadic and low-bandwidth wireless connectivity [34]. For Blockchain under proof of work and its variants, mining requires a lot of computing power and consumes a huge amount of energy. Computationally complex consensus mechanisms are not suitable for IoT scenarios, and the resource needed to reach consensus must be bounded. Proof of stake and its variants are more likely to be used in IoT, but none of these have yet been deployed in IoT as a standard adoption.

C. Privacy

In some sense, users in Blockchain are considered safe, for example, transactions are made with addresses instead of

identities, and users can even generate different addresses for different transactions (which is a practice encouraged in Bitcoin network). However, in distributed ledger, every participant can access every transaction in the network. The identity of a user can be revealed by analyzing address patterns [35], [36]. A more severe problem was reported in [37]: in Bitcoin network, users can be linked to IP addresses, even when these IP addresses are protected by firewalls For a transaction of Blockchain to be validated and confirmed, the transactions must be broadcast to the entire network. This makes it very difficult to preserve privacy. As stated before, privacy is a critical concern in IoT. Taking smart home for example [20], large amounts of safety-critical data and privacy-sensitive information can be generated, processed, and exchanged between device. So the issue of privacy is a critical consideration while deploying Blockchain/DLT in IoT.

D. Predictability

IoT requires predictability. Take time predictability for example. As stated in [38], devices in IoT need real-time interaction with their environment. This means the time used by interactions between things should be predictable and the latency of communication between devices should be bounded. Predictability is even more critical when it comes to health care applications based on IoT [39]. Authors in [40] proposed the concept of real-time Internet of Things (RT-IoT) to highlight the real-time requirements in many IoT applications.

However, Blockchain is not designed with predictability in mind, and consensus is reached in an uncertain manner. For example, the transaction finality in Blockchain under many consensus mechanisms, such as proof of work, proof of stake, is probabilistic [41], and the confirmation confidence of the transaction in tangle [42] is also probabilistic. It remains a fundamental challenge to incorporate predictability concerns in the Blockchain architecture.

IV. Lottery-based Consensus Mechanism

For most of the challenges mentioned above, we can trace their roots back to the consensus mechanisms used in Blockchain or distributed ledger. Over years, lots of consensus mechanisms, such as practical Byzantine fault tolerance, proof of work, proof of stake, have been proposed to address these issues and challenges. But now, It is still not clear which mechanisms are suitable for IoT applications. In the following two sections, we review the major consensus mechanisms and point out their strength and weaknesses when applied in an IoT setting. Hopefully, this review will provide a guide for practitioners while choosing and designing consensus mechanisms for Blockchain/DLT deployed in IoT.

This section focuses on consensus mechanisms used in public distributed ledger, most of which elect a validator through some form of lottery, and Section V focuses on consensus mechanisms used in private or permissioned distributed ledger.

A. Proof of Work

Proof of work is the consensus mechanism used in Bitcoin [2]. In proof of work, a prover demonstrates to a verifier that he has performed a certain amount of computational work in a specified interval of time [43]. Proof of work can be used to deter denial of service attacks and other service abuses such as spam on a network. In Bitcoin protocol, the hashcash function [44] is used and the work consists of computing a hash of a block. The prover adjusts a nonce in a block such that the output hash is lower than or equal to a certain target value. The work is designed to be difficult to solve for the prover and trivial to verify for the verifier.

In proof of work, mining and the longest chain rule is the core of reaching consensus in Bitcoin protocol and practically solving the double spending problem. Mining is the process of adding transaction records to Bitcoins public ledger [45]. The ledger consisting past transactions is called the Blockchain, shown in Figure 1. Every block in this chain mainly contains a hash to the previous block and a list of transactions. The consensus reached by the network in the process of mining is the total order of confirmed transactions, which are stored in the Blockchain. By tracing transactions in the chain of blocks, nodes in the network are able to find the balances of nodes and distinguish legitimate transactions from those attempting to double spend. Mining is deliberately designed to be computation-intensive for two reasons. First, this ensures the block time to be 10 minutes on the average, which can prevent too many forks of the Blockchain. Second, this makes tampering with the Blockchain computationally impractical for one node, which prevents double spending and secures the network.

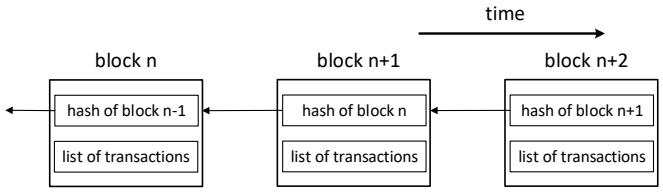

Fig. 1: Simplified Blockchain data structure

The only feasible method to solve the cryptographic puzzle, i.e., to do the proof of work, is to guess the nonce in a block and compute the hash value until the hash satisfies the predefined condition, so a lot of computing power and energy are consumed. Two types of mining reward are provided in Bitcoin protocol: additional Bitcoins and transaction fees. The first miner who discover a block is rewarded a certain number of Bitcoins. Additionally, the miner is awarded the transaction fees, which is the difference between the amount of inputs and outputs of the transaction. As the newly mined Bitcoins decreases over the time, the transaction fees will play a more important role to incentivize participation.

Bitcoin uses a CPU-bound function as the basis for its proof of work scheme [46]. Anyone who possesses a certain amount of computing power is capable of mining blocks and Bitcoins in principal. However, as the continuing development of the Bitcoin community, the mining devices are also changing, from CPU, GPU to FPGA and ASIC. As computation power increase,

the mining difficulty should be adjusted accordingly to ensure a relatively steady block time, which in turn makes it more difficult for small miners to join the mining community. As a consequence, the so-called rich get richer phenomenon [47] appears and the participation democracy is jeopardized.

While proof of work plays an essential role in reaching consensus in Bitcoin, a fundamental criticism of proof of work in general and Bitcoin in particular is that it wastes computational power (and thus energy). Proof of work is subject to the majority attack, or "51% attack", where nodes controlling more than 50% of the network's computing power can reverse transactions that were already confirmed and can double spend coins [48].

B. Proof of Stake and Its Variants

1) Proof of Stake: Proof of stake is an energy-saving alternative to proof of work. In proof of stake, the participants stake determines their likelihood to discover the next block. In the original version of proof of stake [49], the stake is coin age, which is defined as currency amount times holding period. In proof of stake, a new type of transaction called coinstake transaction is introduced. Through coinstake transaction, a miner consumes its stake, such as coin age, and gets paid as a reward for its mining. The reward is proportional to the consumed stake.

The mechanism to reach consensus in proof of stake is similar to proof of work. The fork with highest total consumed coin age is chosen as the main chain. In proof of stake, the miners are also stakeholders, which contrasts with the situation in proof of work, where the capability of mining is related to computing power. Thus, under proof of stake, it is to the miners benefit to maintain consensus and prevent the network from attacks.

Proof of stake is costless, energy-saving compared to proof of work. In proof of stake, the hash is (except for a timestamp) calculated on static data, there is no way for miners to use their computational power to solve the puzzle faster than others. In particular, there is no nonce which can be modified. Instead, every second the timestamp changes and miners have a new chance of finding the solution. However, the low mining cost can lead to the issue of "nothing at stake" [50], where nodes mine on multiple forks to maximize rewards while losing nothing. Consequently, using proof of stake alone is prone to attacks, such as bribe attacks [51].

Proof of work might affect network security with block reward declining over time due to tragedy of the commons [52], and proof of stake is an approach of changing the miner's incentives in favor of higher network security.

In proof of stake, the issue of monopoly typically appearing in proof of work is largely mitigated. The voting power under proof of stake is not only related to the coin amount that miners possess, but also related to time. A poor miner can wait a relatively long time to increase his probability to discover a new block. Subsequently, a relatively equal distribution of voting power is achieved in proof of stake, and "poor get richer" replaces "rich get richer", meaning that every node in proof of stake can validate transactions, thus help enhancing network security.

Under proof of stake, nodes with more than 50% of stake can mount an attack, and double spend their coins. But this is highly impossible in proof of stake. First, an attack has to possess or purchase more than half of the currency in the network, and the result is the cost is considerably higher than the gain. Second, miners are also stakeholders. If an attacker possessing a large number of coins mounts an attack, the confidence of participants in the network will decrease, thus depreciation of the cryptocurrency will ensue, which is to the detriment of the attacker.

A large portion of altcoins use proof of stake, such as Peercoin [49], 2Give [53], and 808Coin [54].

2) Leased Proof of Stake: In proof of stake, although the calculation of voting power takes time into account(holders can waiting for longer time to increase their voting power), it is still unlikely for participants with small balances to validate a block, which is similar to the situation in proof of work. This means only a small portion of larger holder have the right to validate a block, which undermines mining democracy and endangers network security. The more participants, the securer the network. Incentivizing smaller holders to participate is critical for network security. Leased proof of stake [55] achieves this by allowing holders to lease their balances to other stakeholders. The voting power of the staking nodes (nodes who validate blocks) is increased by the leased coins, Subsequently, the chance of validating a block for these nodes is increased. Since mining rewards are shared among the leasers, small holders can be incentivized to participate. This is the approach taken by Waves [56].

Leased proof of stake is designed to incentivize small stakeholders to take part to reach consensus, thus improving network security compared to proof of stake.

3) Delegated Proof of Stake: To solve the same issue facing leased proof of stake, a similar but different approach is delegated proof of stake [57]. With delegated proof of stake, a list of block producers (also called *witness*, the nodes responsible for validating blocks) are elected by nodes in the network, according to their account balances. The major difference between proof of stake and delegated proof of stake is that the former is a direct democratic while the latter is representative democratic. This means in delegated proof of stake, most participants do not create block directly, instead, they vote for witnesses. Unlike leased proof of stake, rewards in delegated proof of stake are given to the witness, not shared among voters. However, stakeholders can compete to become a witness.

According to the white paper of delegated proof of stake [57], like proof of work, the general rule is the longest chain wins. Witnesses take turns to produce a block every 3 seconds. Because the mechanism requires 2/3 + 1 block producers to reach consensus, if more than 1/3 of the block producers are malicious or malfunction, the network will fail. But recall that the witnesses are elected by stakeholders, if they are not

eligible, stakeholders would eventually vote to replace these witnesses.

In delegated proof of stake, with significantly fewer nodes to validate a block, the block can be confirmed quickly, making the transaction latency relatively low. Note that although delegated proof of stake is discussed in this section, it reaches consensus not by lottery, but by voting among witnesses.

4) Proof of Stake Velocity: Under proof of stake, coin age accumulates even when the node is not connected to the network. The lack of a sufficient number of online nodes can facilitate attacks. To promote more active network participation, based on proof of stake, proof of stake velocity [58] is designed to encourage both ownership (stake) and activity (velocity). Different from proof of stake, under proof of stake velocity, a non-linear coin-aging function is introduced. Depending on specific implementations, coin age usually increases quickly at first after a transaction related to this coin was confirmed, and increases become slow over time. The change of coin-aging function can change the incentives and encourage uses to stay online and participate to secure the network.

Proof of stake velocity is still vulnerable to 51% attack. The author of [58] mentioned that due to a non-linear coin-aging function, people cannot simply wait for a long time to increase their stake, thus rendering the difficulty of 51% attack under proof of stake velocity being significantly increased.

C. Consensuses Requiring Specialized Hardware

1) Proof of Elapsed Time: Proof of elapsed time (PoET) [59] consensus mechanism offers a solution to the Byzantine generals problem by utilizing a trusted execution environment (TEE) to improve the efficiency of present solutions such as proof of work. Its approach is based on a guaranteed wait time provided through the TEE. PoET works as follows: Every node requests a wait time from TEE. The node with the shortest wait time for a particular block is elected to validate the block, and thus receives the corresponding reward.

does not require high power consumption or specialized hardware. Participation in PoET requires a CPU with trusted execution environment and does not require high power consumption and specialized hardware required in proof of work. However, one major criticism is also related to TEE, which may be not open and under the control of a particular organization. Besides, in [60], the authors showed that the PoET design is vulnerable in the sense that adversary can jeopardize the Blockchain system by only compromising $\Theta(\frac{\log \log n}{\log n})$ fraction of the participating nodes, which is very small when n is relatively large.

2) Proof of Luck: Proof of luck [61] is a consensus algorithm, which is based on the use of trusted execution environments, and which achieves low transaction latency while using minimal energy and computing power. Under proof of luck, nodes request a random number (luck) from TEE, and a node with the highest luck is elected to validate a block.

In [61], the authors proved that as long as the population size of the attackers is less than a half, the probability of the success of the attack decreases exponentially in the number of blocks, h, after a fork. Same as PoET, a disadvantage of proof of luck is that it requires specialized hardware.

D. Other Consensus Mechanisms

1) Proof of Activity: In a distributed network, the more active nodes, the securer the network. Based on this observation, to solve the incentive problem in proof of work, proof of activity [62] was proposed. Proof of activity is a combination of proof of work and proof of stake. In proof of activity, to validate a block, proof of work should be done in the first place. The first node who solves the cryptographic puzzle signs a block. Then, it comes to proof of stake. A function called *follow-the-satoshi* is used to choose a group of validators. The probability of a node being chosen is proportional to the balances of nodes. After all validators sign the block, the block is validated, and consensus is reached. Same as proof of work, the longest chain wins. Mining rewards and transaction fees are shared between the miner and the validators who sign the block.

Since a group of nodes are chosen to mine the block, the consensus mechanism is complex. In [62], the authors proved that under the assumption that the function, which generates inputs of *follow-the-satoshi*, is a random oracle, an attacker with x fraction of the online stake needs to have more than $(\frac{1}{x} - 1)^N$ times the hashpower of the honest miners in order to gain an advantage over the network, where N is the number of nodes chosen to mine a block. Under the same assumption, the authors prove that if p fraction of the honest stake is online, an attacker with y fraction of the total stake needs more than $((\frac{1}{y} - 1) \times p)^N$ times the hashpower of the honest miners in order to gain an advantage over the network.

Proof of activity is complex and needs proof of work, which is expensive for IoT devices.

2) Proof of Importance: Under proof of stake, the rich can get richer. In order to achieve a more even wealth distribution, proof of importance [63] was proposed. Under proof of importance, each account is assigned an importance score. Accounts with higher importance scores have higher probabilities of harvesting a block (mining a block). A user's importance score is determined by how many coins they have and the number of transactions made to and from their wallet. In order to achieve consensus, Proof of importance utilizes a list of measures, including the topology of the transaction graph, the NCDawareRank [63] network centrality measure. Nodes with more than a certain number of *vested coins* can be eligible for importance calculation, and thus are eligible to validate a block. Vested coins are a portion of the total coins an account holds. That the transaction graph can be used for calculating the importance of an account is the key feature of proof of importance.

Under proof of importance, a score is calculated with respect to a block. A score of a Blockchain is the sum of blocks it includes. The fork with the highest score wins.

The block creation takes into account many factors in the network. No formal analysis is made concerning the majority attack. The authors of [63] mentioned some measures in the

mechanism to counter Sybil attack [64] where a single user generates multiple entities to influence the consensus process.

3) Proof of Space (Capacity): Proof of space [65] is an alternative approach to proof of work, where a service requestor must dedicate a significant amount of disk space as opposed to computation. By this way, huge energy consumption involved in proof of work can be saved in proof of space. A proof of space is a piece of data that a prover sends to a verifier to prove that the prover has reserved a certain amount of space [66]. Under proof of space, a large data set, *plots* are generated by the verifier in advance. Plots are stored on the hard drive of the prover. The amount of plots stored are proportional to the probability of being selected to validate a block. One way of implementing proof of space is by using hard-to-pebble graphs [67]. During the verification, firstly, a prover is asked to build a labeling of a hard-to-pebble graph; secondly, the prover will be required to open several random locations of the graph to show that the information is known for the prover.

Hard drive space should be used to mine. Participation in mining under proof of space depends on disk space and is resistant to ASIC. Proof of space requires hard drive space, while IoT devices usually have limited storage.

4) Proof of Space Time: In proof of space, the disk space can be reused, which can result in the cost per proof being arbitrarily low. To address this issue, under proof of space time (PoST) [68], a prover is required to convince a verifier that the prover spends a space-time resource (storing data over a period of time). Reaching consensus in PoST consists of two phases: an initialization phase and an execution phase. Compared to a proof of work, PoST requires less energy, while time and storage should be used to validate a block.

In [68], the authors presented the definitions of completeness and soundness of PoST, and gave the formal definition of PoST.

5) Proof of Burn: Proof of burn [69] is a solution to the drawbacks of proof of work mining. The idea behind proof of burn is to burn coins, thus reducing energy wasting in proof of work. Proof of burn joins with proof of work and proof of stake to provide block generation and network security. Burning coins means sending your coins to a special address whose coins cannot be spent according to cryptographic mechanisms. When burning coins, a transaction is made to the burn address. Once a burn transaction has been confirmed, the so-called burn hash can be calculated. If the burn hash is lower than a predefined target, then a proof of burn block is discovered. Proof of burn can be used as a viable tool for migration from one cryptocurrency to another.

Coins of another cryptocurrency, usually proof of work based cryptocurrency, are consumed to mine the proof of burn block. To mine, one has to burn coins first. Under proof of burn, voting power can be dominated by those who are willing to burn more coins, which can undermine a wide participation of nodes in the network.

Proof of burn depends on proof of work mechanism. The author of [69] proposed a new hash algorithm for its underlying proof of work mechanism and claimed that the algorithm is to prevent an ASIC dominated proof of work mining scheme from occurring.

Proof of burn mechanism is still susceptible to 51% attack. What makes things complex is proof of burn system is usually a combination of proof of burn, proof of work, and proof of stake. A node possessing 51% hash power is able to attack the system, but it is not clear what the hash power exactly is and how to quantify it.

6) Tangle: Tangle [42] is a distributed ledger based on a directed acyclic graph. Although Blockchain and Tangle adopt different data structures to store transactions, the consensus mechanisms of them are closely connected. Under the CAP theorem [70], Blockchain favors consistency over availability, while Tangle favors availability over consistency. Tangle is with IoT in mind at its design time, and provides features required by IoT industry, such as high scalability and no transaction fees.

A Tangle-based network functions in the following way. The transactions issued by nodes (also called participants or users) constitute the Tangle graph. Transactions are the vertices of the graph. When a node issues a transaction, it must approve two *tips*, and this relationship of approving between transactions constitutes the edge set of Tangle. Tips are unapproved transactions, which are transactions in Tangle without any transactions referencing it. An example Tangle graph is shown in Figure 2. In Tangle network, the following steps are required to issue a transaction: 1) signing: using a private key to sign the transaction; 2) tip selection: using a weighted random walk to select two tips in Tangle to be referenced by the transaction. At the same time, the newly issued transaction also approves the two tips; 3) proof of work: In order to have the transaction accepted by the network, participants need to do some proof of work. After the three steps, the node can broadcast the transaction to the network. The transaction can be selected in tip selection process while other nodes of the network are to issue a transaction, and then, this transaction can be approved and validated.

Similar to Bitcoin, The consensus of Tangle is probabilistic: a confirmation confidence with respect to a transaction is given, which is an indication of its acceptance level. In order to know the confirmation confidence for a particular transaction, a node can perform the tip selection algorithm for certain times, for example, 100 times. Then the percentage of tips, out of the 100 selected tips, which reference the transaction in question is the confirmation confidence. For example, if it is referenced by 80 tips of 100, confirmation confidence of the transaction is 80%.

In Tangle, there are no miners, and nodes who issue transactions are required to approve other transactions. Consequently, no transaction fees are needed, which makes Tangle suitable for conducting micropayment in IoT applications. Tangle favors availability over consistency and is able to achieve high transaction throughput by parallelizing validations. Experiments showed confirmed transactions per second are above 100 in smaller networks of less than 250 nodes, with confirmation time within 10 seconds [71].

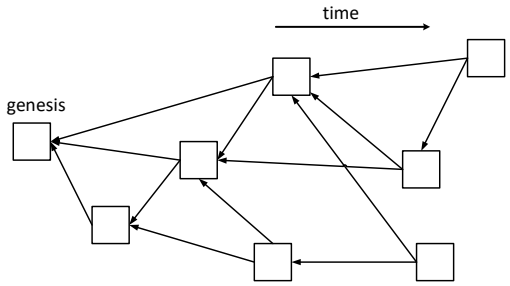

Fig. 2: Tangle data structure

In Tangle network, in order to issue a transaction, proof of work is needed. proof of work prevents an adversary from spamming the network [72]. This means resource wasting, which is undesirable in IoT.

V. Voting-based Consensus Mechanism

Practical Byzantine Fault Tolerance and its variants is reviewed in this section. These consensus mechanisms typically depend on explicit voting in multiple rounds, suffer from issues related to scalability, and are mainly deployed in private or permissioned distributed ledger.

A. Practical Byzantine Fault Tolerance

Practical Byzantine fault tolerance (PBFT) [73] is a replication algorithm that is able to tolerate Byzantine faults. PBFT is designed to reach consensus in distributed asynchronous environments like the Internet under the assumption that at most $\lfloor \frac{n-1}{3} \rfloor$ out of a total of n nodes are simultaneously faulty. Under PBFT, a new block is determined in a round. In each round, a primary, who is responsible for ordering the transaction, will be selected. The whole process can be divided into three phase: pre-prepared, prepared and commit. Through these phases, the total order of the requests can be determined even under the situation where the primary is faulty, and the consensus can be reached. PBFT requires that every node is known to the network.

With a known list of participants in place, under PBFT, consensus can be reached with low network communications and low transaction latency, provided that up to one-third of nodes in the network are faulty. However, the scalability of PBFT is limited, which means PBFT can only be used in a private or permissioned Blockchain, and is not suitable for IoT settings.

Hyperldeger Fabric [74], Parity [75], Tendermint [76] all use variants of PBFT.

B. Proof of Authority

Proof of authority [77] is a new family of Byzantine fault tolerant consensus algorithms largely used in permissioned Blockchain systems to ensure better performance than traditional PBFT. Proof of authority operates in rounds during which an elected participant acts as mining leader and is in charge of proposing new blocks on which distributed consensus is achieved. It does not depend on nodes solving arbitrarily difficult mathematical problems, but instead uses a set of authorities - nodes that are explicitly allowed to create new blocks and secure the Blockchain. The chain has to be signed by the majority of authorities. Compared to proof of work, under permissioned Blockchain, proof of authority is more secure, less computationally intensive, more predictable, and provides lower transaction latency. In a system with N authorities, the mechanism assumes that at least $N/2+1$ of authorities should be honest.

In [78], a qualitative analysis shows that proof of authority algorithms are not actually suitable for permissioned Blockchain deployed over the Internet, because they do not ensure consistency. Parity [75] is an implementation of proof of authority.

C. Proof of Validation

Proof of validation [79] is a variant of PBFT. Proof of validation takes into account the stake of validators in the process of reaching consensus and avoids the nothing at stake problem by utilizing punishment. In order to be a validator, participants need to issue a *bond transaction* and have a portion of their coins locked in a *bond deposit*. A validator has voting power equal to the amount of the bonded coins. Once a validator is found to be dishonest, the bonded coins of the dishonest validator will be destroyed. The validation of blocks is done in a round robin manner among validators. A two-phrase voting is needed to successfully commit a block. A block is committed when more than 2/3 of validators vote for the block. Tendermint [76] uses proof of validation to reach consensus.

D. Hashgraph

Hashgraph [80], as a distributed ledger, is an alternative to Blockchain. Unlike Blockchain, which uses a chain of blocks to store data, in Hashgraph network, data are stored in a directed acyclic graph (DAG). Hashgraph consensus algorithm is proposed for replicated state machines with guaranteed Byzantine fault tolerance [80]. The properties of Hashgraph include fairness, asynchrony, no leaders, no round robin, no proof of work, consensus with probability one, and high speed in the absence of faults. Gossip about gossip and virtual voting are two techniques used in Hashgraph to achieve consensus. In Hashgraph network, gossip protocol is used to propagate information, and the information being spread is the history of the gossip itself, i.e. the Hashgraph, which is why it is named gossip about gossip. After the process of gossip about gossip, the Hashgraph is spread across the network, and every node sees a copy of the Hashgraph, although, due to asynchronous network, different nodes may see different Hashgraphs. Consequently, nodes of the network can calculate votes of other nodes and reach consensus without the actual votes being spread through the network.

Hashgraph mechanism works in the following manner: nodes in the network send *events* to a randomly chosen neighbor. The information in an event mainly includes transactions, the hashes of its two parent events. Events are vertices of Hashgraph. The two parents events are the latest event of the sending node and

the latest event of its randomly chosen neighbor, and these relations between events constitute the edge set of Hashgraph. The events can spread across the network and reach each node in a relatively fast manner, because of the rapid convergence of gossip protocol. The history of the gossip process can be illustrated by a directed acyclic graph, as shown in Figure 3. In Figure 3, the network contains three nodes, and the circles represent events. New events are appended to the right side of the graph. After the information spreading using gossip protocol, virtual voting can be conducted. If a transaction is confirmed by over two-thirds of nodes in the network, it will be seen as valid.

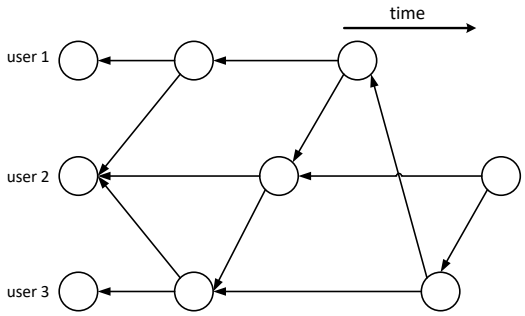

Fig. 3: Hashgraph data structure

Hashgraph can achieve low transaction latency and high transaction throughput, which is limited only by the network bandwidth. According to [81], in a network with 32 members, Hashgraph can process around 250,000 transactions with latency less than one second. Hashgraph is not resource wasting, in the sense of avoiding proof of work.

However, note that at this time, Hashgraph is only deployed in private, permissioned networks [82]. In a public network like Bitcoin network, since nodes are allowed to join and leave at any time, no list of nodes is known beforehand, and no trust exists among nodes. Consensus mechanisms in public settings have to take measures against maliciousness, such as Sybil attack. While in permissioned network, arbitrary participation in the network is forbidden, and list of nodes is known beforehand and there is trust among nodes. This difference can lead to a degradation of performance while deploying Hashgraph in a public IoT network. The scalability of Hashgraph may be a critical issue to address.

VI. Conclusion

Internet of Things (IoT) makes a worldwide network of interconnected devices and human beings possible, but still faces many serious challenges. Distributed ledger technology (DLT), with many attractive features, such as decentralization, persistency, anonymity, and auditability, not only brings tremendous opportunities for new application and business models, but also appears to be promising to address these technical challenges. IoT connects things together, and DLT offers an architecture to organize things. We believe the convergence of these two trends will bring a significant revolution on not only the technology level and also the society level in the future. In this paper, we discuss the benefits and challenges while deploying Blockchain/DLT in IoT, review the existing consensus mechanisms on a distributed ledger, and summarize their pros and cons with respect to their potentials for the IoT environment.

Acknowledgment

This work is supported by the RGC of Hong Kong (ECS 25204216 and GRF 15204917), UGC of Hong Kong (Project 1-ZVJ2), the National Natural Science Foundation of China (Grant No. 61772123, 61672140 and 61532007) and the Ministry of Education Fund Project (Grant No. 6141A020333).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

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

[3] "The internet of trusted things - kaleido insights," http://www.kaleidoinsights.com/reports/internet-of-trusted-things-blockchain/, (Accessed on 03/25/2018).

[4] A. Tapscott and D. Tapscott, "How blockchain is changing finance," *Harvard Business Review*, vol. 1, 2017.

[5] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*. IEEE, 2016, pp. 1–3.

[6] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*. IEEE, 2015, pp. 131–138.

[7] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Work Pap.–2016*, 2016.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016.

[10] F. d. O. Sergio, J. F. da Silva Junior, and F. M. de Alencar, "The blockchain-based internet of things development: Initiatives and challenges," *ICSEA 2017*, p. 39, 2017.

[11] J. Sun, J. Yan, and K. Z. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financial Innovation*, vol. 2, no. 1, p. 26, 2016.

[12] P. Brody and V. Pureswaran, "Device democracy: Saving the future of the internet of things," *IBM, September*, 2014.

[13] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[14] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[15] A. Norta, "Creation of smart-contracting collaborations for decentralized autonomous organizations," in *International Conference on Business Informatics Research*. Springer, 2015, pp. 3–17.

[16] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, and D. Boyle, *From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.

[17] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for iot data," in *Web Services (ICWS), 2017 IEEE International Conference on*. IEEE, 2017, pp. 468–475.

[18] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 173–178.

[19] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of iot data," in *Proceedings of the 2017 on Cloud Computing Security Workshop*. ACM, 2017, pp. 45–50.

[20] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 618–623.

[21] S. Kumar, "Not just for cryptocash: How blockchain tech could help secure iot," *IoT Agenda*, vol. 13, 2017.

[22] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "openpds: Protecting the privacy of metadata through safeanswers," *PloS one*, vol. 9, no. 7, p. e98790, 2014.

[23] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and privacy," *arXiv preprint arXiv:1712.02969*, 2017.

[24] G. Zyskind, O. Nathan *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 180–184.

[25] "Etherapis: Decentralized, anonymous, trustless apis," https://etherapis.io/, (Accessed on 03/31/2018).

[26] "True micropayments with bitcoin - earn.com - medium," https://medium.com/@earndotcom/true-micropayments-with-bitcoin-e64fec23ffd8, (Accessed on 03/31/2018).

[27] "filecoin.pdf," https://filecoin.io/filecoin.pdf, (Accessed on 03/31/2018).

[28] "The future of energy — blockchain, transactive grids, microgrids, energy trading — lo3 stock, tokens and information — lo3 energy," https://lo3energy.com/, (Accessed on 03/31/2018).

[29] "Irreversible transactions - bitcoin wiki," https://en.bitcoin.it/wiki/Irreversible_Transactions, (Accessed on 04/03/2018).

[30] "Confirmation - bitcoin wiki," https://en.bitcoin.it/wiki/Confirmation, (Accessed on 04/10/2018).

[31] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology? -a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[32] "Machinomy - distributed platform for iot micropayments," https://machinomy.com/documentation/vision-paper/, (Accessed on 04/03/2018).

[33] "Blockchain size - blockchain," https://blockchain.info/charts/blocks-size, (Accessed on 04/03/2018).

[34] P. Danzi, A. E. Kalør, Č. Stefanović, and P. Popovski, "Analysis of the communication traffic for blockchain synchronization of iot devices," *arXiv preprint arXiv:1711.00540*, 2017.

[35] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 127–140.

[36] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 6–24.

[37] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 15–29.

[38] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[39] N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*. ACM, 2011, p. 131.

[40] C.-Y. Chen, M. Hasan, and S. Mohan, "Securing real-time internet-of-things," *arXiv preprint arXiv:1705.08489*, 2017.

[41] A. Baliga, "Understanding blockchain consensus models," Tech. rep., Persistent Systems Ltd, Tech. Rep, Tech. Rep., 2017.

[42] S. Popov, "The tangle," *cit. on*, p. 131, 2016.

[43] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*. Springer, 1999, pp. 258–272.

[44] A. Back *et al.*, "Hashcash-a denial of service counter-measure," 2002.

[45] "Mining - bitcoin wiki," https://en.bitcoin.it/wiki/Mining, (Accessed on 04/10/2018).

[46] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[47] D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, "Do the rich get richer? an empirical analysis of the bitcoin transaction network," *PloS one*, vol. 9, no. 2, p. e86197, 2014.

[48] "51% attack — investopedia," https://www.investopedia.com/terms/1/51-attack.asp, (Accessed on 04/08/2018).

[49] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[50] "Problems - ethereum/wiki wiki - github," https://github.com/ethereum/wiki/wiki/Problems, (Accessed on 04/08/2018).

[51] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 142–157.

[52] G. Hardin, "The tragedy of the commons," *Journal of Natural Resources Policy Research*, vol. 1, no. 3, pp. 243–253, 2009.

[53] "2give," http://2give.info/, (Accessed on 03/16/2018).

[54] "Proof-of-stake coins list !" https://www.poslist.org/, (Accessed on 03/16/2018).

[55] "Waves launches balance leasing in lite client - waves platform," https://blog.wavesplatform.com/waves-launches-balance-leasing-in-lite-client-14db9eac0377, (Accessed on 03/16/2018).

[56] "Waves platform," https://blog.wavesplatform.com/, (Accessed on 03/16/2018).

[57] "Dpos consensus algorithm - the missing white paper - steemit," https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper, (Accessed on 03/16/2018).

[58] L. Ren, "Proof of stake velocity: Building the social currency of the digital age," *Self-published white paper*, 2014.

[59] "Poet 1.0 specification - sawtooth v1.0.1 documentation," https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture/poet.html, (Accessed on 03/16/2018).

[60] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (poet)," in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*. Springer, 2017, pp. 282–297.

[61] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: an efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*. ACM, 2016, p. 2.

[62] I. B. C. L. A. Mizrahi and M. Rosenfeld, "Proof of activity: Extending bitcoins proof of work via proof of stake," 2014.

[63] "Nem_techref.pdf," https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf, (Accessed on 03/16/2018).

[64] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

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

[66] "Proof-of-space - wikipedia," https://en.wikipedia.org/wiki/Proof-of-space, (Accessed on 04/11/2018).

[67] C. Dwork, M. Naor, and H. Wee, "Pebbling and proofs of work," in *Annual International Cryptology Conference*. Springer, 2005, pp. 37–54.

[68] T. Moran and I. Orlov, "Rational proofs of space-time," 2017.

[69] "slimcoin_whitepaper.pdf," http://www.doc.ic.ac.uk/ ids/realdotdot/ crypto_papers_etc_worth_reading/proof_of_burn/slimcoin_whitepaper.pdf, (Accessed on 03/16/2018).

[70] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *Acm Sigact News*, vol. 33, no. 2, pp. 51–59, 2002.

[71] "A primer on iota (with presentation)-iota," https://blog.iota.org/a-primer-on-iota-with-presentation-e0a6eb2cc621, (Accessed on 04/05/2018).

[72] Q. Bramas, "The stability and the security of the tangle," 2018.

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

[74] "Hyperledger fabric - hyperledger," https://www.hyperledger.org/projects/fabric, (Accessed on 03/16/2018).

[75] "Parity," https://www.parity.io/, (Accessed on 03/16/2018).

[76] "Tendermint - blockchain consensus," https://tendermint.com/, (Accessed on 03/16/2018).

[77] "Proof of authority chains - paritytech/parity wiki - github," https://github.com/paritytech/parity/wiki/Proof-of-Authority-Chains, (Accessed on 03/16/2018).

[78] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2017.

[79] J. Kwon, "Tendermint: Consensus without mining," *Retrieved May*, vol. 18, p. 2017, 2014.

[80] L. Baird, "Hashgraph consensus: fair, fast, byzantine fault tolerance," Swirlds Tech Report, Tech. Rep., 2016.

[81] "hh-whitepaper-v1.0-180313.pdf," https://s3.amazonaws.com/hedera-hashgraph/hh-whitepaper-v1.0-180313.pdf, (Accessed on 04/04/2018).

[82] "Demystifying hashgraph: Benefits and challenges - hacker noon," https://hackernoon.com/demystifying-hashgraph-benefits-and-challenges-d605e5c0cee5, (Accessed on 04/11/2018).