Review article

A survey on consensus methods in blockchain for resource-constrained IoT networks☆

Mehrdad Salimitari*, Mainak Chatterjee, Yaser P. Fallah

Department of Computer Science, University of Central Florida, Orlando, FL 32825, United States

A R T I C L E   I N F O

A B S T R A C T

The success of blockchain as the underlying technology for cryptocurrencies has opened up possibilities for its use in other application domains as well. The main advantages of blockchain for its potential use in other areas are its inherent security mechanisms and immunity to data manipulation attacks. A blockchain relies on a consensus method for agreeing on any new data. Most of the consensus methods which are currently used for the blockchain of different cryptocurrencies require high computational power and thus are not suitable for resource-constrained systems.

In this article, we survey the various blockchain-based consensus methods that are applicable to resource-constrained IoT devices and networks. In a typical IoT network, there exist several devices with limited computational and communication capabilities. Most often, these devices cannot perform intensive computations and are starved for bandwidth. Therefore, we discuss the possible measures that can be taken to reduce the computational power and convergence time for the underlying consensus methods. We also talk about some of the alternatives to the public blockchain, such as private blockchain and tangle, along with their potential adoption for IoT networks. Furthermore, we review the existing consensus methods that have been implemented and explore the possibility of utilizing them to realize a blockchain-based IoT network. Some of the open research challenges including AI-enabled blockchains are also put forward.

Published by Elsevier B.V.

1. Introduction

We are witnessing the proliferation of Internet of Thing (IoT) applications in our houses, offices, neighborhoods and cities. Adoption of IoT based technologies are poised to make a big impact on various sectors of our daily lives, including energy, manufacturing, intelligent transportation and smart cities. With more and more autonomous deployments of potentially large-scale IoT systems, ensuring security, availability and confidentiality of the data, the devices and the networks become utmost critical [1,2].

In order to realize an entirely autonomous IoT network, different sensors and devices (generically referred to as 'nodes') in an IoT network need to communicate with each other in a distributed fashion. This requires a mechanism by which different nodes in an IoT network can agree upon the validity of any communicated data. One of the best ways to achieve this is to use a consensus method. There exist different consensus methods by which different nodes can reach a com-

mon decision without participation of a central controller [3]. Consensus methods typically require high computational and communication capabilities. There are some that are less demanding, however, they do not provide strict guarantees on the performance attributes [4].

As far as the computational and communication capabilities of the IoT devices are concerned, they are typically inexpensive low-powered embedded computing platforms capable of performing their bare minimum tasks. Most embedded IoT devices are equipped with 8-bit or 16-bit microcontrollers with very little RAM and storage capacities, and can connect to the Internet either via ethernet or low-powered wireless communications such as IEEE 802.15.4 [5]. These resource constraints make it difficult to directly apply the traditional consensus protocols to IoT networks. However, newer technologies like blockchain [4] and tangle [6], that use consensus methods for accepting new data, have the potential to be customized depending on the requirements and constraints of various IoT devices and networks [7].

This article reviews how blockchain works and argues that full-fledged implementations of blockchain (as in bitcoin) are impractical for resource-constrained IoT networks. Though blockchain usually employs very compute-intensive hash operations for cryptocurrencies, such operations are not essential for non-critical systems as resource-constrained systems might be willing to trade some level of data integrity for savings in computations and energy consumption. Private blockchains, that allow participation of valid users and are not open to the public, have been proposed to overcome this problem. These blockchain implementations use consensus methods that do not require high computational power for solving hash problems. However, since they are private, only valid users can access them. We compare the pros and cons of various consensus protocols and blockchain implementations alongside their applicability towards IoT networks. We highlight some of the research challenges that need to be addressed for widespread deployment of blockchain-based IoT networks.

The rest of the article is organized as follows. In Section 2, we briefly discuss what blockchain is. In Section 3, we survey what the requirements of IoT networks and also what the limitations of the current technologies are. In Section 4, we motivate how blockchain can be applied to IoT networks for addressing the current limitations of IoT networks. In Section 5, we study different consensus methods. In Section 6, we survey the existing blockchain implementations and discuss their pros and cons. In Section 7, we put forward some of the open research challenges. Conclusions are drawn in the last section.

2. An overview of blockchain

Blockchain is a distributed and immutable ledger made out of unalterable sequence of blocks [4,8,9]. Blockchain topology can be categorized into three groups: public, private, and consortium [10]. The main difference among these categories is how the users can access the ledger and participate in the consensus [11]. Consensus protocol is the backbone of any blockchain network that enables the immutability of ledger and distributed access. In addition, there exist some other distributed ledger topologies that share the same characteristics as blockchain but have a different architecture. Directed acyclic graph (DAG) is the most famous blockchain alternative that is used in Tangle consensus protocol [12]. It is discussed in details in Section 5.7.

2.1. Public blockchain

A public blockchain is a distributed and tamper-resistant database that no single entity controls, but can be shared and accessed by all. New records (called blocks) can be added to the existing blocks as long as the new block is approved by all in the network. Also, once blocks are recorded, it is not feasible to modify or erase them [13]. Blockchains have been designed to work on an unreliable network with adversarial entities. By using sophisticated and compute-intensive secure hash algorithms, blockchain achieves data integrity which prevents data erasure, data manipulation, and recording invalid information [10,14,15]. These compute-intensive algorithms are part of *proof of work* which is a method of consensus by which different nodes in a network can agree upon new data or detect an anomaly. However, there exist other consensus methods with significantly lower computational requirements and network overhead. In Section 5, we review proof of work and other consensus methods in details.

Blockchain stores all data transactions in chained blocks, i.e., a block contains several transactions linked to previous blocks by a hash pointer [16]. This chain continues to the first block mined in blockchain which in bitcoin blockchain was mined by *Satoshi Nakamoto* and is called the *Genesis block* [17]. This architecture makes it infeasible to modify the transactions in the blockchain because for changing a transaction, it is required to change all the blocks built on that transaction before a new block is added to the blockchain. This modification might be feasible only with more than half of the hash rate power of the world.

Fig. 1 illustrates a blockchain of 4 blocks each of which contains the `Block number`, a (random number) `Nonce`, the `data`, the hash of the previous block (i.e., `Prev`) and the hash of the current block (i.e., `Hash`). Since there is no block prior to block 1, the previous hash is set to all 0ʼs. With the block number, data and previous hash known, the objective is to find the nonce such that the hash of all is less than a target value, i.e., SHA-256 (`Block#, Nonce, Data, Prev`) ≤ `target value`. Suppose the nonce found is 42,345 as shown in block 1. The target value indicates the difficulty level of the proof of work. In other words, this target value mandates that the 256-bit output of the hash function starts with a certain number of consecutive zeros. In the illustration, all hashes are shown to start with three 0ʼs. For block 2, the same process is repeated to obtain the nonce with the new block number and the data. Also, `Prev` is updated with the `Hash` of the previous block.
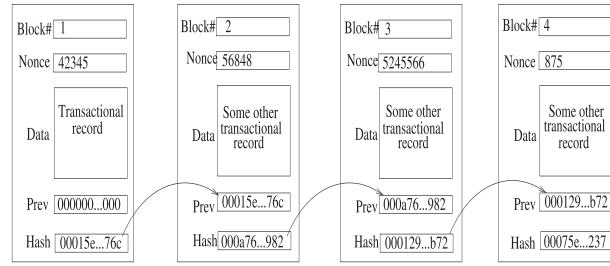
Fig. 1. Chaining of blocks in blockchain.

Again, we observe that a different nonce is obtained and the Hash begins with three consecutive 0′s. It is to be noted, as the target value decreases, the difficulty of finding the nonce increases.

2.2. Private blockchain

There exist some blockchains designed by different companies for specific applications with restricted access to the public [18]. Having access to these blockchains or participating in their consensus protocol is permissioned and dependent on a third-party. Therefore, private blockchains are centralized to some extent which is in contradiction to the original idea of blockchain being totally distributed [11]. However, these blockchains have lower computational requirements and faster network response time which make them more desirable for IoT applications [12]. In some private blockchains, some of the nodes have complete access to all the stored blocks in the blockchains. Thus, private blockchains bring more privacy to the information which is desirable for companies [4].

Private blockchains are more secure than traditional databases because of using cryptographic protocols similar to public blockchains. However, they are not as secure as public blockchains that employ computational-intensive protocols like proof of work. Thus, there is the possibility of tampering stored data in private blockchains [4]. However, restricted access of users to the consensus protocol prevents data tampering in private blockchains. Private blockchains can follow different consensus methods like practical Byzantine fault tolerance, proof of elapsed time and proof of stake which are discussed in Section 5.

2.3. Consortium blockchain

There exist some partially private blockchains called consortium blockchains. While a fully private blockchain is controlled by a single company, consortium blockchains are governed by several institutions all of which directly participate in the consensus protocol [13]. These institutions are responsible for maintaining the regulations regarding how the users can participate in the consensus protocol. They also indicate the level of accessibility of different users to the ledger. Similar to private blockchains, users can be granted access to the entire ledger or only part of it according to the user's credentials. The most known consortium blockchains are part of the Hyperledger project which is a collaboration between many well-known companies and hosted by the Linux foundation [19]. Consortium blockchains can employ different consensus methods similar to private blockchains. Since the only difference between the architecture of the private and consortium blockchains is the number of governing institutions, they are both referred as private blockchain throughout the paper.

3. IoT: Importance and limitations

We are witnessing the use of IoT networks in various domestic, industrial and military applications. A common feature of these IoT networks is containing several sensors and actuators which are resource-constrained devices capable of communication without human intervention. Besides these devices, there are other network entities that connect the sensors and actuators to the backbone network infrastructure. These are routers, switches, aggregators and cloud infrastructure comprising virtual servers and storage– all of which dictate the baseline requirements for resource provisioning and sharing. These requirements include dynamic and verifiable group membership of devices, authentication and data integrity, robustness against single point of failure, lightweight operations in terms of resources and low latency communication [20].

3.1. Client-server paradigm deficiencies

Current IoT solutions mostly rely on the centralized client-server paradigm where all the devices are identified, authenticated and connected via cloud servers [21]. Therefore, these servers require enormous amount of processing capability and storage capacity. Besides, all the communications between the devices have to go through the Internet even if the devices are close to each other. Although these models are practical for small IoT networks, they do not scale well. Furthermore, the cost of establishing multitude of communication links, maintaining centralized clouds and networking all equipment, is remarkable for large-scale IoT networks. In addition, reliance on cloud servers makes the architecture susceptible to single

point of failure which reduces the robustness. Moreover, IoT devices must be immune to manipulation attacks and physical tampering. Though, there exist some methods trying to make the IoT devices secure, they are complex and not appropriate for resource-constrained IoT devices with limited computational power [20].

Considering the growth rate of IoT networks and mentioned shortcomings of client-server models, there is a demand for newer methods to make the network more decentralized in the future [22]. One of the suggested solutions is the creation of large peer-to-peer (P2P) wireless sensor networks. However, it does not also adequately address the privacy and security requirements of IoT networks [23–25]. The deficiencies of existing technologies for large-scale implementation are discussed in more details next.

3.2. Limitations of existing IoT technologies

Following are some of the main challenges for large-scale deployments of existing IoT implementations.

- *High cost:* Current IoT Technologies usually require centralized clouds and server farms that impose high costs for deployment and maintenance. In addition, these requirements necessitate middlemen which make the infrastructure even more centralized and costly.
- *Maintenance:* Apart from the high cost related to maintenance of millions of smart devices within the network, applying required software updates to all of these devices is a considerable burden.
- *Privacy:* For extensive deployment of IoT networks, users need to be assured about the privacy and anonymity of their data. Some companies permit certain authorities (e.g., governments, manufacturers or service providers) to access and control users' devices.
- *Security:* Existing IoT solutions use closed-source codes which do not provide transparency to users. Future IoT implementations would require utilization of open-source codes so that all the users can maintain and update the software making it less susceptible to malicious activities. Furthermore, current IoT technologies are mostly prone to single point of failure; as a result, if a cloud server confronts a physical or software malfunction, it may impact the entire network functionality [26]. Also, in client-server paradigm, a breach in a single device connected to a server or cloud can wreak havoc with the entire system through denial of service (DoS) attacks, sending malicious messages to other devices, leaking private data, or manipulating the aggregate data [27–30]. Collected data from different devices is stored, processed and forwarded by intermediate systems which can tamper the data. In addition, wireless channels used for data broadcast are unreliable which makes the network prone to jamming attacks [12]. More detailed analysis of possible security breaches in IoT networks are examined in [31] and [32].

On the contrary, decentralized network of blockchain can easily address some of these limitations like high cost and maintenance. In addition, different consensus methods used in various blockchain implementations can improve privacy and security deficiencies of current IoT networks. The potential of different consensus methods for addressing these issues is surveyed in Section 5.

4. Blockchain-based IoT networks

The applicability of blockchain to IoT networks and its promises for addressing existing IoT technologies deficiencies are discussed in this section. Given the stringent requirements of IoT networks, blockchain looks very apt for both i) securing the stored data in the network against manipulation attacks and ii) providing a secure platform for all devices in the network to communicate with each other without requiring a trusted authority or central server [17,33].

4.1. Smart home: An example of IoT networks

One of the first uses of IoTs is the automation of homes and cities which are referred as *smart homes* and *smart cities*. A smart home involves creating a local network consisting of various smart devices, sensors, and meters within a house. This network is made accessible from anywhere to the residents of the house through the Internet. One of the main obstacles towards realizing this goal is securing the communicated and stored data in this network against malicious acts that desire to wreak havoc with someone's home.

In a smart home network, there is a significant amount of data that is exchanged among different nodes. Electricity meters need to access all the electrical appliances in the house. It has to interact with the smart car for charging and discharging (during peak times, one might use the stored energy in the car as a backup for electricity demands). The HVAC system needs to communicate with the thermostat and weather forecasting systems. Similarly, the fire extinguisher receives signals from smoke detectors. All these connections, including the residents need to remotely access the devices, require exchange of data in a secure manner.

In addition, in a smart home, there is a need to store all the communicated data among the devices and the current state of each device and sensor for analysis and better decision making. For example, historical data along with resident's power consumption habits can be used for energy saving decisions and recommendations. Also, some smart homes can learn and adapt the device parameters based on historical observations. In such cases, system logs are very instrumental in accurate

learning, model building, predictive analysis, and decision making [34]. Needless to emphasize, the stored data needs to be immune to data manipulation attacks.

Extending the concept of smart homes, one can imagine of a *smart city* as a large network connecting various smart homes, smart transportation systems, smart power grids, smart vehicles, and so on. Securing such a large network is a bigger concern since any vulnerability in any of the smaller networks can be exploited to launch bigger attacks with significant damages due to cascading effects. Compared to a smart home, a smart city's requirements are more stringent and demands even a more secure communication links and data storage/exchange protocols.

Blockchain has emerged as probably the most promising technology to immunize communication links and stored data in IoT networks such as smart homes. The communicated data in blockchain is immutable because of its underlying consensus protocol. Thus, manipulation attacks on transmitted or stored data are not plausible through a single compromised node and majority of nodes ought to be compromised for a successful attack [35].

4.2. Blockchain and IoT integration

Blockchain establishes a peer-to-peer network which distributes the computational and storage requirements among all the devices within the network. Thus, it eliminates the cost of installation and maintenance of centralized clouds, data centers and networking equipment. This communication paradigm solves the single point of failure problem as well by decentralizing the network. In addition, blockchain solves the privacy issues for IoT networks by using cryptographic algorithms [15]. It also addresses the reliability concerns in IoT networks by using tamper-resistant ledgers [20].

Despite the built-in mechanisms that guarantee data integrity in blockchain, its implementation for resource-constrained IoT networks is challenging due to the following reasons. First, solving the hash puzzles as an essential part of many consensus methods is compute-intensive and demands immense CPU cycles. Second, the communication links could become a bottleneck in delivering numerous transactions to all the devices and getting their approvals in every second. The problem is even more aggravated in an interference-limited wireless system when all IoT devices have to vie for shared radio links. Third, IoT networks contain many devices that need to communicate with each other fast and at all times. This necessitates adding many blocks to the blockchain every second which requires low latency consensus methods [36].

Resource-constrained IoT networks usually have some flexibility over their performance requirements and are ready to trade some level of data integrity for savings in computations and energy consumption. One solution to achieve that is to relax the proof of work in order to reduce computational requirements. Another way is avoid maintaining all the blocks since as the number of blocks increase, more storage is required. Rather, the last *l* blocks could be chained and withdraw the old blocks to facilitate the storage requirements. Although such approaches would not yield the same level of data integrity as cryptocurrencies, they would nevertheless guarantee enough data protection for IoT networks. However, a more feasible solution to empower resource-constrained IoT networks with blockchain is using novel consensus methods with significantly lower computational requirements, network overhead and latency. These methods are discussed next.

5. Consensus methods

Consensus methods are the essential part of any blockchain network by which different nodes agree on a new block to be added to the blockchain. The consensus method is what enables a blockchain network to function in a distributed fashion. A blockchain-based system is as secure and robust as its underlying consensus method. The most well-known consensus method is proof of work which is used by bitcoin. However, during the past years, a multitude of consensus methods are designed and developed to be used for different applications. We present existing consensus methods and discuss the possibility of applying them to a blockchain-based IoT network. In addition, we compare all the discussed consensus protocols in Table 1 to indicate the most promising ones for IoT networks. The consensus protocols that are not applicable to IoT networks or are mere modifications of a general consensus method are discussed in brief.

5.1. Proof of work (PoW)

Proof of work is a computationally expensive consensus approach. In this method, different nodes in a blockchain try to solve a cryptographic hash function, SHA-256 in particular. SHA-256 generates a unique, fixed size 256-bit hash. The process of finding the next block through solving this hash problem is called *mining* and the users performing this task are called *miners*. The solved block contains transactions, hash of the previous block, nonce and a time stamp. After a miner finds a valid nonce value which solves this problem, he releases his solved block so other miners can be aware of it. In the next step, other miners verify the claimed block by using the block information and its claimed associated nonce as an input to a SHA-256 function. If its output is less than the target value, the miner will accept it as a valid block and withdraw all of his effort for solving that block and move on to find the next block [35]. The miners' incentive for verifying a block is to avoid spending time and computational resources on an already solved block. On the other hand, it is vital for miners to be the one who finds the next block while validating a new block. What withholds miners from accepting a new block without validating it and mining the next block is that whenever the blockchain detects an invalid block, it will discard that block and all the blocks built upon it. A block gets more credibility when more blocks are built upon it. This is

Table 1
Comparisons of different consensus methods [18,50,51,55,61,67,68,71,72].

Consensus method	Access[a]	Decentralization	Scalability	Latency[b]	Throughput[c]	Adversary tolerance	Computing	Network overhead	Storage overhead	IoT suitability
PoW	Public, PL.	High	High	High	Low	<25% Computing Power	High	Low	High	○
PoC	Public, PL.	High	High	High	Low	<50% Storage Space	Low	Low	Very High	○
PoET	Private, P. or PL.	Medium	High	Low	High	N/A	Low	Low	High	●
PoS	Public P. or PL.	High	High	Medium	Low	<51% Stakes	Medium	Low	High	◐
DPoS	Public, PL.	Medium	High	Medium	High	<51% Validators	Medium	N/A	High	◐
LPoS	Public PL.	High	High	Medium	Low	<51% Stakes	Medium	Low	High	○
PoI	Public PL.	High	High	Medium	High	<51% Importance	Low	Low	High	◐
PoA	Public, PL.	High	High	Medium	Low	<51% Online Stakes	High	Low	High	○
Casper	Public, PL.	High	High	Medium	Medium	<51% Validators	Medium	Low	High	○
PoB	Public, PL.	High	High	High	Low	<25% Computing Power	Medium	Low	High	○
PBFT	Private, P.	Medium	Low	Low	High	<33% Faulty Replicas	Low	High	High	●
dPBFT	Private, P.	Medium	High	Medium	High	<33% Faulty Replicas	Low	High	High	◐
Stellar	Public, PL.	High	High	Medium	High	Variable	Low	Medium	High	◐
Ripple	Public P.	High	High	Medium	High	<20% Faulty UNL nodes	Low	Medium	High	◐
Tendermint	Private, P.	Medium	High	Low	High	<33% Voting power	Low	High	High	◐
ByzCoin	Public, PL.	High	High	Medium	High	<33% Faulty Replicas	High	Medium	High	○

Table 1 (*continued*)

Consensus method	Access[a]	Decentralization	Scalability	Latency[b]	Throughput[c]	Adversary tolerance	Computing	Network overhead	Storage overhead	IoT suitability
Algorand	Public, PL.	High	High	Medium	Medium	<33% Weighted Users	Low	High	High	◯
Dfinity	Public, P. or PL.	High	High	Medium	N/A	N/A	Low	N/A	N/A	◯
RSCoin	Private, P.	Low	High	Low	High	N/A	Low	Medium	High	◯
Elastico	Public, PL.	High	High	High	Low	<25% Faulty VAlidators	Medium	High	High	◯
OmniLedger	Public, PL.	High	High	Medium	High	<25% Faulty Validators	Medium	Medium	Low	◑
RapidChain	Public, PL.	High	High	Medium	High	<33% Faulty Validators	Medium	Low	Low	◑
Raft	Private, P.	Medium	High	Low	High	<50% Crash Fault	Low	N/A	High	◑
Tangle	Public, PL.	Medium	High	Low	High	<33% Computing Power	Low	Low	Low	●

[a] PL. denotes permissionless and P. denotes permissioned.
[b] High latency is in order of minutes, medium latency is in order of seconds, and low latency is on order of milliseconds.
[c] Low throughput is less than 100 TPS (transaction per second), medium throughput is between 100 TPS and 1000 TPS, and high throughput is more than 1000 TPS.

known as confirmation in cryptocurrencies' blockchains. After several confirmations, a block and its contained transactions are considered to be acceptable.

In order to add a block in PoW, the miner should find a specific nonce value for each block in a way that the hash value of that block be less than a specific target value. This process is mathematically hard and time consuming because it can only be done by brute force search and the miners have to try different nonce values randomly to reach the target value [35]. It should be noted that this target value designates the network difficulty. For bitcoin, the target value is defined such that the mining process takes 10 minutes in average. Adversaries can affect PoW-based blockchains such as bitcoin by gaining control over 25% of computational power via an attack known as selfish mining [37]. Although this attack cannot jeopardize the immutability of bitcoin's blockchain, it can increase revenue of adversaries.

Although Proof of work has proved to be an effective approach for cryptocurrencies over years, it does not seem to be practical for IoT networks due to its high computational and bandwidth requirements. There exist some other consensus methods which are based on proof of work. These methods have tried to address some of the limitations of PoW specially its high computational requirement. These methods are discussed next.

5.1.1. Proof of capacity (PoC)

Proof of capacity is similar to PoW but instead of depending on the computing power of the miners, it relies on their hard disk capacity. Thus, it is significantly more energy-efficient than ASICs mining used in PoW. In PoC, miners have to store huge data sets, known as plots, to get an opportunity to mine the next block. Therefore, by storing more plots, a miner will gain a higher chance to solve the next block [3]. The block creation time in PoC is 4 minutes. PermaCoin and SpaceMint are two cryptocurrencies employing PoC. Aside from the high latency, this method is not a rational choice for IoT networks where the devices have limited storage capacity.

5.1.2. Proof of elapsed time (PoET)

Proof of elapsed time is a consensus method proposed by Intel which works similar to PoW but with significantly lower energy consumption. In this method, miners have to solve a hash problem similar to that of PoW. However, instead of a competition between miners to solve the next block, the winning miner is randomly chosen based on a random wait time. The winning miner is the one whose timer expires first. The verification of correctness of timer execution is done using a trusted execution environment (TEE) like Intel's software guard extension (SGX) [38].

PoET's eased computational requirements make it IoT friendly. In addition, its low latency and high throughput make it favorable for IoT networks. The main drawback of this approach is its dependency on Intel which is in conflict with the basic philosophy of blockchain being entirely decentralized.

5.2. Proof of stake (PoS)

After proof of work, this is the most prevalent consensus method used for cryptocurrencies. This method works similar to proof of work but with a significant difference. It does not induce a race amongst the nodes to solve the next block. Alternatively, a node is chosen by lottery to mine the next block based on its proportional stake in the network, i.e, its wealth in terms of that cryptocurrency. The selected node will use a digital signature to prove its ownership over the stake instead of solving a complicated hash problem. As a result, it does not require high computational power. In this method, all the coins (i.e., the cryptocurrency) are available from the first day and no mining reward or coin creation exists and the miners are rewarded only with a transaction fee [3].

Although this method eliminates the computational requirements of proof of work, it creates new problems. This method is contingent upon nodes with the highest amount of stakes which makes the blockchain somewhat centralized. Furthermore, there is another problem called "nothing at stake" which refers to the situation in which a selected node has nothing to lose if it behaves badly. Therefore, nothing prevents a node from, for example, creating two sets of new blocks to obtain more reward from transaction fees. However, some modifications are applied to this method to deal with these problems. Although, it has significantly eased computational requirements of proof of work, it is not yet popular for resource-constrained IoT networks. In addition, this method and its variants (which are discussed below) are based on monetary concepts (stakes) that does not exist in IoT networks.

5.2.1. Delegated proof of stake (DPoS)

This method is based on the proof of stake consensus method. Contrary to PoS which is direct democratic, this method is representative democratic [4] which means that all the stakeholders vote to choose some nodes as *witnesses* and *delegates*. Witnesses are responsible and rewarded for creating new blocks. The delegates are responsible for maintaining the network and proposing changes such as block sizes, transaction fees, or reward amount. In each election round, *N* witnesses with the highest votes are chosen. *N* is defined such that at least 50% of the voting stakeholders believe there is enough decentralization. It should be noted that the number of witnesses that each stakeholder vote for, must be at least equal to their desired number of witnesses for decentralization. In DPoS, there are built-in mechanisms in place for detecting and voting out a malicious delegate or witness [3,39,40].

Bitshares, a cryptocurrency, uses DPoS which has shown to significantly improve throughput and latency compared to PoS, but at the cost of making the blockchain more centralized. It is capable of processing 100,000 transactions per second

(TPS). In addition, a block is added to the blockchain in 1.5 s on average and in 3 s maximum [3,39,40]. Though these features make DPoS very attractive for IoT networks, the main bottleneck for DPoS in IoT networks is its dependency on monetary concepts (stakes) to choose witnesses and delegates.

5.2.2. Leased proof of stake (LPoS)

It works the same way as PoS but with some improvements. LPoS tries to solve the centrality problem in PoS. It enables the nodes with low balances to participate in block verification by adding a leasing option. Leasing allows the wealth holders with higher balances to lease their funds for specific amount of time to nodes with low balances. The leased amount will be in possession of the wealth holders during the lease contract, however, it will increase the chance of solving a block for nodes with low balances. When these nodes solve a block, they will share the reward with the wealth holders proportionally. This approach makes blockchain more secure by making it more decentralized [41]. Unfortunately, LPoS is not useful in the context of IoT since it is based on monetary concepts which is not necessarily applicable to an IoT network.

5.2.3. Proof of importance (PoI)

PoI is a modified version of PoS where instead of considering only nodes' balances to determine the next winning node for solving the next block, it takes into account more factors including a node's reputation which is specified by a particular system defined function and the number of transactions occurred to or from that node. Therefore, this consensus method considers productive network activity of nodes which is more efficient than only nodes' balances [42]. The cryptocurrency NEM uses PoI for consensus. PoI yields high throughout and exhibits comparatively low latency. In addition, it does not require large computational or network overheads. These features make PoI favorable for IoT networks. However, PoI like other PoS-based methods is dependent on monetary concepts.

5.2.4. Proof of activity (PoA)

Proof of activity is a hybrid consensus method based on proof of work and proof of stake [4]. First, miners try to solve a hash function in a competition to find the next block as in proof of work. However, the solved block will only contain a header and the miner's address without any transaction. Then, transactions are added to the block and according to the solved block's header, a group of validators is chosen to sign the new block in order to reach consensus. This step is done by using proof of stake. This approach is safer against attacks but can experience higher delay which might not be acceptable for delay-sensitive IoT applications [3].

5.2.5. Casper

Casper is a PoS-based consensus protocol for transitioning ethereum's consensus method from PoW to PoS. It is an overlay on the existing PoW blockchain in ethereum [43]. It prevents "nothing at stake" attacks by slashing all the attacker's stakes. A modification of the *greedy heaviest observed subtree (GHOST)* is used as the chain selection rule in Casper. GHOST is proposed as an alternative for the longest-chain rule used in the bitcoin blockchain. In blockchain networks, there is a possibility for forks because of the network delay. In those cases, the longest chain is selected as the main chain and other branches are ignored. In the GHOST protocol, at each fork, the heaviest subtree rooted at the fork is selected in the chain [44,45]. This method provides significant benefits for cryptocurrencies in terms of security and delay. However, they are not sufficient for tackling the IoT challenges in comparison with the naive PoS method.

5.2.6. Proof of burn (PoB)

Proof of burn is based on burning coins which refers to sending coins to an irretrievable address. Miners get priority to solve the next block according to the amount of coins they have burnt [3]. While this approach is practical for designing cryptocurrencies, it is not appropriate for IoT applications since it is contingent upon existence of a monetary framework and burning of coins, neither of which is inherent in an IoT network. A cryptocurrency called Slimcoin uses PoB.

5.3. Byzantine agreement methods

These consensus methods are based on Byzantine generals problem. In short, this problem is to come up with a consensus method between Byzantine generals over the attack strategy with the assumption that some generals may be traitors and try to adopt treacherous actions to prevent loyal generals reach a consensus and conquer the city. The scenario for this problem is illustrated in Fig. 2. The following are the Byzantine agreement-based consensus methods.

5.3.1. Practical byzantine fault tolerance (PBFT)

In this method, all the nodes should participate in the voting process in order to add the next block and the consensus is reached when more than two-thirds of all nodes agree upon that block. PBFT can tolerate malicious behavior from up to one-third of all nodes to perform normally. For instance, in a system with one malicious node, there should be at least 4 nodes to reach a correct consensus. Otherwise, consensus is not reached. In this method, the consensus is reached quicker and more economically compared to proof of work. Also, it does not require owning assets similar to proof of stake to take part in the consensus process [3].
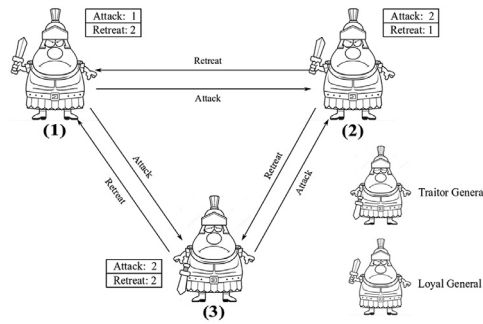
Fig. 2. Byzantine problem with 3 generals. PBFT can only tolerate malicious activity by less than 1/3 of nodes. In this example, general 3 is traitor who can prevent the loyal generals to a reach consensus in case they have different opinions by sending them a different decision from their own decisions. Here, loyal general 1 decides to attack and loyal general 2 decides to retreat. Traitor general 3 uses the arisen opportunity from loyal generals' conflict of opinions and sends them two different messages containing a decision in contrast with their own decisions. Therefore, based on the received messages, general 1 decides to retreat and general 2 decides to attack.

This method is well-suited for private blockchains like the hyperledger projects that are controlled by a third-party. However, it is not the best choice for permissionless, public blockchains due to their limited scalability and comparatively low tolerance towards malicious activities. PBFT has high throughput, low latency, and low computational overhead– all of which are desirable for IoT networks. However, its high network overhead makes it un-scalable for large networks, thus it could be applied only to small IoT networks.

5.3.2. Delegated byzantine fault tolerance (dBFT)

dBFT follows the same rules as PBFT but it does not require participation of all the nodes for adding a block which makes it more scalable. In dBFT, some nodes are chosen as delegates of other nodes and according to some rules, they pursue the consensus protocol similar to PBFT [4]. A cryptocurrency called NEO uses this consensus method. This method has many desirable features similar to that of PBFT. However, its average latency for block creation is 15 s which is not acceptable for an IoT network.

5.3.3. Stellar consensus protocol (SCP)

Stellar Consensus Protocol provides micro-finance services on the blockchain platform. It was proposed by Mazieres using a variant of PBFT called federated Byzantine fault tolerance (FBFT) as the backbone [46]. In FBFT, nodes belonging to inter-secting groups (i.e., the federates) run a local consensus protocol among their members [47]. This method is decentralized and is open to the public which allows everyone to participate in the consensus protocol. It has a very low latency similar to web transactions (a few seconds at most). This is the first Byzantine agreement-based consensus method which provides users with the maximum freedom to choose among different combinations of other participants to trust for consensus.

SCP reaches robustness through quorum slices. A quorum is a set of nodes that participate in the consensus protocol and a quorum slice is its subset that helps a node in its agreement process. Individual trust decisions made by participants in the blockchain constructs a quorum slice, and quorum slices connect the whole network together in a similar fashion as peering networks create the Internet by binding together. It should be noted that the quorums are selected by the nodes involved in the transactions.

SCP consists of two steps: nomination protocol and ballot protocol. First, nomination protocol is executed. During this step, new values called candidate values are proposed for agreement. These values are sent to all the nodes in the quorum and each node will vote for a single value among the candidate values. At the end of process, unanimously values are chosen for that slot. Then, ballot protocol is initiated which involves a federated voting to either accept or abort the obtained values in the nomination protocol. Finally, the ballot for the current slot is finalized and aborted ballots are discarded. In situations that nodes cannot reach a consensus to abort or to commit a value, a higher valued ballot is initiated which can be considered as a new ballot protocol execution [46,48].

High throughput and low computational requirements of this method are desirable for IoT networks. Although its latency is comparatively low, it is not in order of milliseconds which is required for IoT networks. If the latency of this method could be decreased by reducing the network overhead (which is $O(n)$), this method can be a good choice for IoT networks.

5.3.4. Ripple

Similar to stellar, ripple uses FBFT consensus method. It is proposed to reduce the latency of blockchains [49]. In this decentralized method, each miner uses a trusted subset of nodes within the larger network to reach a consensus. There are two types of nodes in the network: server nodes which are responsible for the consensus protocol and client nodes which only transfer funds. Each server node contains a unique node list (UNL). Nodes within an UNL are used for reaching consensus over new transactions. When 80% of the nodes within an UNL agree over a transaction, the consensus is reached.

The ripple consensus protocol is executed every few seconds by all the nodes in order to reach consensus over new transactions. Ripple can tolerate up to 20% faulty nodes in the UNL [50]. This method is mostly used for monetary purposes to enable transactions with no chargebacks [51]. It has similar features as stellar which makes it a good candidate for IoT networks if its latency is decreased to order of milliseconds.

5.3.5. Tendermint

Tendermint belongs to a family of Byzantine fault tolerance (BFT) consensus protocols which can host arbitrary application states [52]. It is a permissioned consensus method [50]. Contrary to PBFT where each node has equal voting power, in Tendermint nodes have different voting powers that are proportional to their stakes [47]. Therefore, it can be considered as a hybrid consensus method based on PBFT and PoS. Tendermint can tolerate malicious activity from at most one-third of the total Byzantine voting power [53].

Participants in the tendermint protocol are called validators which propose blocks of transactions and vote on them in turn. The voting process consists of two steps: pre-vote and pre-commit. A block is added to the blockchain when more than two-thirds of validators pre-commit for the same block in the same round [54]. Unlike PBFT, validators are chosen based on PoS by locking their coins and dishonest validators are punished [55]. Thus, this method relies on monetary concepts like PoS which is not necessarily applicable to IoT networks. Considering high scalability, high throughput, and low latency, this method could be applied to IoT networks if the monetary concept is replaced by some other criteria.

5.3.6. Byzcoin

ByzCoin is a Byzantine consensus that uses a collective signing protocol called CoSi [56] to commit bitcoin transactions within seconds which makes PBFT scalable. It also modifies PBFT to enable a public blockchain by supporting dynamic membership proportional to PoW as in bitcoin. This method uses a tree-structured communication protocol which significantly reduces the latency of bitcoin's blockchain. However, it is vulnerable to DoS attacks [57].

This method yields a throughput higher than the PayPal network with a confirmation latency of 15–20 s [57]. Although scalability and throughput of this method is desirable for IoT networks, its latency is beyond the acceptable limits of IoT networks. Furthermore, the PoW used in this method is similar to that of bitcoin [57] which is not appropriate for IoT devices.

5.4. VRF-based methods

In verifiable random function (VRF)-based methods, committee members are randomly selected for participation in the consensus protocol [58]. Algorand and Dfinity are two such methods.

5.4.1. Algorand

Algorand is a public and permissionless blockchain implementation that uses pure proof of stake (PPoS) consensus protocol built on Byzantine agreement [59]. Algorand has been proposed as a new cryptocurrency to address limitations of existing implementations: decentralization, scalability, and security. Most of the available blockchain implementations are centralized to some degree; e.g., in a bitcoin network, the users with the highest hash power control the network. However, in Algorand, each block is approved by a unique committee of users that are randomly selected by VRFs in a private and non-interactive fashion using the users' private key and public information from the blockchain. Thus, it is fully decentralized. This selection is based on users' weights assigned according to the amount of money in their account. This step is based on proof of stake [60].

After the committee members are selected, the committee reaches a consensus over the new block using a Byzantine agreement protocol called $BA\star$. Existing Byzantine fault tolerance protocols need a fixed set of servers to be determined ahead of time which makes them vulnerable to Sybil attacks, where an adversary creates many pseudonyms to affect the Byzantine agreement protocol. In addition, they do not scale to a large number of users. However, $BA\star$ eliminates the risk of Sybil attacks and can scale to millions of users because of the existing first step in Algorand. $BA\star$ can tolerate up to 1/3 of weighted users (1/3 of the money owned by users) to be malicious like any other BFT based methods. However, because of using randomly selected VRFs in the first step, adversaries do not know which users to attack until they start participating in $BA\star$ which significantly increases the security of the consensus protocol [60].

The main drawback of Algorand is that the randomness used in producing the VRF seeds can be biased by an adversary. To overcome this shortcoming, it uses a look-back mechanism to ensure strong synchrony and unbiased seeds. However, this makes the network vulnerable to "nothing at stake" attack [61].

Latency of Algorand is less than a minute with transaction finality of around one minute. This feature is very desirable for a cryptocurrency. For instance, in bitcoin network it takes about 10 minutes to grow a chain by one block and it requires to wait for 6 blocks to ensure transaction finality (transaction be on an authoritative chain) which takes about an hour. There is not also a possibility of fork in Algorand [59,60].

Although Algorand propounds a bright horizon for financial purposes, it lacks several features to be successfully applied to IoT networks. Algorand does not require communication among users to determine if they are selected to participate in the consensus protocol. It also does not postulate computational resources to solve cryptographic puzzles. These characteristics alongside with decentralization, high scalability, and high security are appealing features of Algorand for IoT. However,

an IoT network requires a delay less than a second (in order of milliseconds). Furthermore, there is no monetary concept in an IoT network to define stakes and assign weights.

5.4.2. Dfinity

Dfinity blockchain proposes a four-layer consensus protocol which can be used for permissioned networks or paired with Sybil resistance methods (e.g., proof-of-work or proof-of-stake) for permissionless and public networks. The most important feature of Dfinity is containing decentralized randomness beacon acting as the VRFs to produce random outputs as the seed. This new VRF protocol is based on a unique-deterministic, non-interactive, DKG-friendly threshold signature scheme which solves the biased coin problem and "nothing at stake" problem which exist in Algorand. Dfinity consensus is also immune to selfish mining attacks [62]. Dfinity can reach consensus over a new block in seconds and transaction finality after two confirmations [62]. Similar to Algorand, this high latency is not acceptable for IoT applications.

5.5. Sharding-based methods

Sharding is splitting the overheads of processing transactions among multiple, smaller groups of nodes called shards [61]. These methods were initially proposed to address the scalability problems in blockchain [63]. Shards work in parallel to maximize the performance of the blockchain (processing more transactions in each consensus round) by sharding different overheads of blockchain network including communication, computation, and storage overhead [64]. The stat-of-the-art sharding methods are discussed below.

5.5.1. RSCoin

RSCoin is a cryptocurrency framework proposed for centrally-banked cryptocurrencies to maintain complete control over the monetary supply using distributed set of authorities called mintettes (validator nodes). It uses a sharding-based approach to make traditional banking systems (with centralized monetary supply) more transparent via a distributed network while maintaining the scalability of the network. Danezis and Meiklejohn claim that their framework avoids double-spending [65]. However, its two-phase commit protocol is not Byzantine fault tolerant and is susceptible to double-spending attacks by a colluding adversary [61]. Moreover, this framework is not decentralized as it is asserted since it relies on a trusted source of randomness for sharding.

Although this framework has a very low latency (in the order of milliseconds) and high throughput [65], it is not applicable for IoT networks. It is mostly centralized and based on a centralized monetary supply which has no place in an IoT network.

5.5.2. Elastico

This is the first sharding-based consensus protocol proposed for public and permissionless blockchains that uses a combination of classical Byzantine consensus protocol (e.g., PBFT) and proof of work to improve bitcoin blockchain. It partitions the network into smaller committees which process disjoint set of transactions (shards). The number of committees grows near linearly in the total computational power of the network. Each committee contains a small number of nodes that reach a consensus over their set of transactions (shard) using classical Byzantine consensus. In each consensus round, these committees are selected using PoW's least-significant bits which are obtained by each node solving a PoW puzzle based on the consensus epoch randomness obtained from the last state of the blockchain [66].

Although this method can enhance latency and throughput of the bitcoin blockchain, it confronts several limitations: (i) It still suffers from a large communication overhead which is troublesome for IoT networks. (ii) The randomness used during each consensus epoch can be biased by an adversary which compromises the committee selection process. (iii) While each node is responsible for verifying a subset of transactions (a shard), it still has to broadcast all blocks to other nodes and store the entire ledger which does not address communication overhead and resource-constrained IoT devices. (iv) It can only tolerate up to 1/4 faulty nodes [61].

Unfortunately, Elastico does not seem practical for IoT implementations for several reasons: (i) While improving bitcoin's latency, the latency is still large enough for IoT networks. (ii) It has very large communication overhead. (iii) The storage requirements are not minimalist. (iv) There are still security issues.

5.5.3. OmniLedger

OmniLedger is a permissionless and public blockchain implementation that uses VRFs and sharding with a ByzCoin-based consensus protocol called ByzCoinX. It has been proposed by Kokoris-Kogias et. al. to address some of the limitations of Elastico, specially latency and scalability. ByzCoinX improves the PBFT part of ByzCoin in order to increase its tolerance to DoS attacks. During each consensus epoch, the committee selection process is done by VRFs in a similar fashion as the lottery algorithm of Algorand. These committees reach a consensus over a subset of transaction (a shard). Similar to Elastico, it can tolerate only up to 1/4 faulty nodes. When there are less than 1/8 faulty nodes, this method can reach a comparatively low latency (less than 10 s) [67].

Aside from the undesirable latency of this protocol for IoT networks and its security issues, this method poses a large communication overhead which cannot be implemented on resource-constrained IoT devices. In OmniLedger's consensus protocol, each committee has to communicate with all the nodes for each block of transaction. In addition, there is another
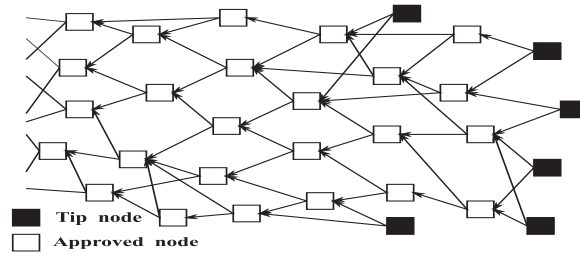
Fig. 3. Tangle protocol.

communication overhead during producing the random seed for the VRF [61]. However, OmniLedger provides low storage overhead since validators are not required to store the full transaction history which is in favor of IoT devices [67].

5.5.4. RapidChain

RapidChain has been proposed to solve existing limitations of sharding-based consensus methods including communication overhead, security, scalability and latency. Unlike other sharding-based methods which have limited tolerance to Byzantine faults, it can tolerate up to 1/3 faulty nodes without any trusted setup. RapidChain is the first public blockchain that provides full sharding: communication, computation, and storage overhead. This method significantly enhances latency and throughput by eliminating the need for linear amount of communication per transaction. This protocol does not require gossiping transactions to the entire network which improves the communication overhead [61].

The full sharding used in this method is very desirable for resource-constrained IoT devices to successfully participate in the consensus protocol. In addition, RapidChain has a high throughput (processing 7,380 transaction per second in a network of 4,000 nodes). However, its latency for confirming the transactions is not suitable for IoT networks (8.7 s for 4,000 nodes). If the latency of this method can be improved, it would be a worthy option for IoT networks [61].

5.6. Raft

Raft is a voting-based consensus method that was proposed to make Paxos algorithm more understandable and implementable for practical systems. Paxos algorithm tries to solve the consistency problem in certain conditions for Byzantine generals problem. Raft achieves the same efficiency as Paxos [52,68]. Raft and Paxos are non-Byzantine fault tolerance algorithms. While their protocols are similar to BFT algorithms, they can only tolerate crash faults up to 50% of the nodes while Byzantine algorithms can tolerate arbitrary (including maliciously) corrupted nodes. It is composed of two stages: leader election and log replication. The leader is responsible for ordering the transactions. The leader selection stage is executed using a randomized timeout for each server when an existing leader fails. After a leader is chosen, log replication stage is triggered. In this stage, the leader accepts log entries from clients and broadcasts transactions to make its version of the transaction log [52,69]. Corda and Quorum are two blockchain implementations that use Raft as their consensus method [47].

This method has high throughput and low latency. However, its performance depends on the leader node which occupies an absolute dominance in the system. Therefore, if the leader node is maliciously infected, the entire system will be destroyed. It cannot tolerate malicious nodes and can endure 50% nodes of crash fault [68]. Since it is crucial to secure the leader node, the throughput is limited by the performance of that node. Due to its low security and restricted throughput, it is not very appropriate for IoT networks.

5.7. Tangle

Tangle is a new technology for distributed ledgers proposed by the cryptocurrency Iota. It does not require a complicated, time consuming and computational-intensive consensus protocol. It also does not use blocks to store transactions. Each transaction is a unique block by itself which must approve two older transactions in order to be added to the ledger. Tangle uses directed acyclic graph (DAG) in which each transaction is linked to two older transactions that are approved by it. After a transaction approves two older transactions, it is added to the ledger through proof of work. The tangle framework is illustrated in Fig. 3. Tip nodes, indicated by solid black squares, are the new transactions waiting for approval. The approved transactions are shown by the white squares. Note, each transaction is linked to two older transactions.

Due to the unique design of tangle, it is a fast, infinitely scalable framework which makes it well-suited for IoT networks. Furthermore, Tangle has no transaction fees which is desirable for an IoT network. In contrast to most of the blockchain implementations, tangle is immune to becoming obsolete with the advent of quantum computers because of its unique design. The main challenge about tangle is how to choose the two older transactions for approval. No rule is imposed by tangle on how to choose these two nodes which is very desirable for resource-constrained devices in an IoT network. However, the chosen transactions should not be the same or conflicting. To choose between conflicting transactions, tangle

runs an algorithm called tip selection algorithm multiple times to figure out which of these two transactions are more plausible to be indirectly approved by the selected tip.

Unlike blockchain frameworks, tangle's design enables parallel transaction verification which eliminates the required wait time for mining previous blocks as in blockchain and provides the opportunity to verify more transactions in a shorter time. Although tangle is very promising and claims to overcome the existing barriers for decentralization of resource-constrained IoT networks [6], it confronts a lot of implementation challenges, specifically for IoT applications. The current implementation of tangle, Iota, does not provide all the claimed goals of tangle. One of the challenges for applying tangle to IoT networks is the storage limitation. The resource-constrained IoT devices are unable to store the entire tangle. Some solutions including automated snapshotting and a swarm client have been proposed to address this problem in Iota's development road map [70]. Another problem with tangle is that whoever gains control over more than one-third hash power of tangle can make it insecure and vulnerable. As a preventive measure, Iota runs a node called 'coordinator' by amassing the hash power itself at one point. However, this can be perceived as centralization of tangle.

5.8. Comparisons of different consensus methods

The above mentioned consensus methods have been used in various blockchain implementations. As the consensus method is the backbone of a blockchain implementation, most of the features and performance attributes of a blockchain-based network are contingent upon its inherent consensus method [68]. Therefore, each consensus method can only address needs of specific applications.

According to the desired application, different requirements should be fulfilled to implement a practical blockchain-based network. The most important features that are considered for employing a consensus method in a blockchain implementation are network accessibility, degree of decentralization, scalability (i.e., size of the network), latency, throughput (number of transactions per second), security (adversary tolerance), computational overhead, network overhead and storage overhead. Regarding the desired application, some of these features become more essential.

In fact, the most appealing application for researchers in blockchain domain is cryptocurrency. Therefore, new consensus protocols are usually proposed based on the cryptocurrency requirements. The most important features for a cryptocurrency are high throughput and security. However, low latency is the most essential feature for IoT networks. In a practical IoT network, a transaction should be sent and finalized within a few milliseconds or less. Unfortunately, most of the existing consensus methods lack low latency and can partially address IoT requirements.

Although low latency is essential for any IoT network, some IoT networks may require more features. For instance, smart home is a network of resource-constrained devices and sensors which require low computational, network, and storage overhead. On the other hand, some IoT networks such as vehicular ad hoc network (VANET) are not resource-constrained and can properly work with more computational, network, or storage overhead. However, scalability becomes an issue for smart transportation networks. While smart homes are small networks composed of tens of devices and sensors, smart cities consist of thousands of devices which require high scalability. In addition, security and network accessibility are important concerns for military-based sensor networks. In Table 1, all the surveyed consensus methods are compared to verify their applicability to IoT networks considering the discussed criteria. Most appropriate consensus methods for IoT networks are denoted with

6. Blockchain implementations

In order to achieve a blockchain-based IoT network, the most appropriate blockchain framework must be designed and deployed. One option is to implement a new framework and use the preferred consensus method which we believe best suits the IoT application requirements. The other option is to use an already defined framework. In this section, we review some of the available frameworks for implementing a blockchain system and their practical applicability towards an IoT application. We survey different aspects of these implementations including their consensus method, their applications, being permissioned or permissionless, being private or public. All these features affect the characteristics of a blockchain framework along with its scalability, performance and availability. Scalability refers to the size of a blockchain network and number of users it can support. Performance refers to the latency and throughput which are critical for IoT networks. Latency is the amount of time it takes for the nodes reach to a consensus. Low latency can be achieved by compromising the decentralization of the blockchain framework. Throughput is the number of transactions that can be processed per time unit. Availability refers to the accessibility of the nodes to a copy of the distributed ledger [4].

The various private blockchain implementations are presented in Section 6.1. One of the well-known frameworks for implementing a private blockchain based system is the Hyperledger framework. Hyperledger is a collaborative project started by the Linux Foundation and developed by many companies including IBM, Intel, Cisco, Hitachi and so on. It contains several projects including fabric, sawtooth, indy, iroha and burrow. As part of the public blockchain based system, we discuss the well-known ones: ethereum and bitcoin in Section 6.2 We also survey the implementations for blockchain alternatives including corda and IoTa in Section 6.3. We compare the suitability of these frameworks for IoT applications in Section 6.4.

6.1. Private blockchains

Hyperledger Fabric: Hyperledger Fabric is a permissioned implementation which is widely used by enterprises [73]. It uses a pluggable consensus method which is defined based on specific application requirements. Its most common consensus method is PBFT. This private blockchain framework attains consensus within hundreds of milliseconds [74]. Such low latency is crucial for building blockchain-based IoT networks. Being permissioned, the blockchain is controlled by a specific organization that allows specific nodes to join the blockchain, access to the database and participate in the consensus protocol. This framework supports `chaincode` designed in the `Go` language which is a special version of smart contracts [75]. Smart contract is a self-executing software written in a programming language that allows users to program their own scripts for transferring financial assets, products, or services between different parties without a middleman [76].

Smart contracts are beneficial for Internet of Things in order to empower different applications. In addition, developers can design different security mechanisms and define different measures to ensure the network privacy using smart contracts. These features become more vital in unmanned IoT networks that the sensors must autonomously perform in a decentralized fashion. Smart contracts can play an important role in ensuring security and privacy of such sensor networks. Unlike financial applications of blockchain that the transactions are all triggered by people, in IoT networks, transactions are usually autonomous and carried out by different sensors. Smart contracts can be employed to manage and process these transactions in an efficient and secure manner [12].

Although low latency of this implementation is a noticeable advantage for IoT networks, it remains a private blockchain and therefore lacks the beneficial features of public blockchains such as being totally distributed with highly secure and immutable data storage. Furthermore, its network overhead significantly increases with increase in the number of nodes because it augments the number of messages communicated in the PBFT protocol. This prevents hyperledger fabric to be used in large-scale applications similar to public blockchains.

Hyperledger fabric contains several entities that maintain the blockchain: peers, orderer, chaincode and chaincode policy. Peers are responsible for validating and performing the requests by the nodes within the network. The validation of new blocks is also the responsibility of the peers. The orderer receives all the valid invoke requests called transactions from nodes, creates a block from of these transactions, sends them to peers to be verified, and adds to their copies of the ledger. These invoke requests are first verified by one or more of the corresponding peers which are dedicated according to the defined policy in the chaincode. Thereafter, these requests are sent to the orderer for further process. Different kinds of messages can be defined on the chaincode by which nodes propagate different requests to peers. The most used messages for a chaincode are query and invoke. Query is getting to know the current state of the ledger and invoke is requesting a change in the ledger [77].

Hyperledger Sawtooth: Proposed by Intel, Hyperledger Sawtooth is a modular platform for implementation of distributed ledgers for storing digital records aptly designed for enterprise usage. It uses proof of elapsed time using Intel's software guard extensions (SGX) as a trusted execution environment for achieving consensus. This platform allows large-scale implementation of both permissioned and permissionless ledgers, and has features such as live data stream, hardware security and enterprise-grade customer load which make it suitable for IoT devices [78]. However, this framework is not yet implemented in a large scale and fully tested for its performance capabilities. It is also not recommended by Intel for security sensitive applications due to its lack of security mechanisms.

6.2. Public blockchains

Bitcoin: The well know cryptocurrency, bitcoin, utilizes a permissionless public blockchain framework. Being permissionless, it allows any node to participate in the consensus protocol and mine blocks without any permission. It uses proof of work as the method for consensus which has a high latency about 10 minutes, making it ineffective for IoT networks. However, it is worth exploring if it can still be used with eased proof of work to reach a consensus in a short time. Alternatively, this framework can be used in combination with other methods. Another significant challenge in applying bitcoin blockchain to IoT networks is its scalability. Average transaction size for bitcoin is between 400 Bytes and 600 Bytes. Considering the size of each bitcoin's block which is 1 MB and the average time to solve each block which is 10 minutes, current bitcoin architecture allows 4 transactions per second which is not acceptable for IoT networks.

Ethereum: Ethereum is a permissionless public blockchain framework developed using solidity (i.e., a contract-oriented, high-level language for implementing smart contracts) [79]. In this implementation, all the nodes are required to participate in the consensus process. It was primarily designed by a derivative of proof of work known as Ethash. This method is significantly less computational-intensive than the original proof of work because of using directed acyclic graph (DAG). This blockchain can be customized and adopted for a variety of applications because of its intrinsic characteristics that enable smart contracts [74]. There are plans for migrating ethereum to Casper [4]. Its block generation process takes between 10 to 20 seconds which is much less than bitcoin's latency but still is not practical for an IoT network implementation [74]. It should be noted that some private and permissioned blockchain frameworks are also designed based on the ethereum blockchain.

Table 2
Comparisons of various blockchain implementations.

Implementation	Hyperledger Fabric	Hyperledger Sawtooth	Bitcoin	Ethereum	Corda	Iota
Consensus method	Pluggable (PBFT generally)	Proof of elapsed time	Proof of Work	Ethash (PoW) Casper (PoS)	Pluggable (Raft generally)	Tangle
Accessibility	Private	Private	Public	Public	Private	Public
Mode of operation	Permissioned	Permissioned or Permissionless	Permissionless	Permissioned or Permissionless	Permissioned	Permissionless
Decentralization	Partially	Partially	Yes	Yes	Partially	Partially
Compute-intensive	No	No	Yes	Partially	No	No
Network-intensive	Yes	No	No	No	No	No
Scalability	Low	High	High	High	Partially	High
Throughput	High	High	Very low	Low	High	High
Latency	100 ms	Very Low	10 Minutes	12 Seconds	Very Low Not Measured	10 ms
Immutability	Low	Low	High	High	High	High
Adversary tolerance	33.33% Faulty Replicas	Unverified	<25% Computing Power	<51% Stakes	unverified	33.33% Computing Power
Privacy	High	High	Low	Low	High	Low
Smart contract	Yes	Yes	Limited	Yes	Yes	No
Currency	None but Tokens possible	None but Tokens possible	Bitcoin (BTC)	Ether (ETH), Tokens possible	None	Iota

6.3. Blockchain alternatives

<u>Corda:</u> Corda is a permissioned decentralized ledger framework that uses pluggable consensus method [80]. In this implementation, specific nodes called notary nodes are responsible for the consensus protocol. Due to the requirement of trusting the notaries for consensus, it is partially decentralized. It cannot be considered as a blockchain framework since it uses a different architecture. This framework is specifically designed for financial applications and not very suitable for resource-constrained IoT networks [79].

<u>Iota:</u> Iota is a distributed ledger that uses directed acyclic graph (DAG) instead of a blockchain. This protocol is called tangle which was discussed in Section 5.7. Its low latency is very desirable for IoT applications and is the first cryptocurrency which has been specifically designed for IoT applications. This design minimizes the transaction time and network overhead at the cost of relaxing security with some possible attack scenarios [74]. Iota does not have any transaction fee which enables micro transactions. Unlike other cryptocurrencies that use a number of confirmations as an indicator for the reliability of a transaction, Iota uses cumulative weight which is defined for each transaction based on the number of consecutive linked transactions to it [6]. In Iota, all the tokens have been made available from the first day and they were released by the first node called the genesis transaction.

6.4. Comparisons of various implementations

In order to assess which implementation addresses most of the limitations of blockchain and is applicable to IoT networks, we compare them in Table 2. Similar to the comparison we made for different consensus methods in Table 1, the same features are noteworthy for comparing different blockchain implementation. The only difference is that blockchain implementation may offer smart contracts or special cryptocurrency. A desirable blockchain implementation for IoT networks should have the following features: decentralized, not compute-intensive, not network-intensive, high scalability, high throughput, very low latency and preserving privacy. However, as was discussed in Section 5.8, existing consensus methods and blockchain implementations can only provide some of these desirable features. It depends on our IoT application to choose the most apt blockchain platform for our network.

The mentioned features in Table 2 for hyperledger sawtooth are claimed in theory and they are not yet fully tested in a large-scale implementation under different circumstances. In addition, hyperledger sawtooth is still under experiment and not recommended by Intel to be used for security sensitive applications. High throughput and low latency of hyperledger fabric is in small-scale implementations and for large-scale networks they will deteriorate. The latency values in Table 2 are average values. Bitcoin and ethereum have high scalability only with very limited network throughput. In Corda framework, a new transaction is dependent on many other transactions and if a node has not seen older transactions formerly will result in a significant high latency which pose some restrictions on scalability. However, generally, its latency is very low. Furthermore, it is claimed in corda's white paper that it is secure against adversarial attacks which is not tested and implemented yet.

7. Directions for future research

Considering the unique features of blockchain, it can be applied to a number of domains including but not limited to IoT networks, healthcare, data storage, inventory tracking and finance. The primary challenge is how to adapt the blockchain

technology to suit the specific application needs. As every application poses different requirements, a new or a customized implementation of blockchain is needed.

7.1. Research challenges for blockchain-based IoT networks

In the IoT domain, the major research challenges that ought to be addressed are:

- *Enhancing scalability:* Scalability refers to the size of a blockchain network and the number of users it can support. In a practical IoT network, a large number of devices need to communicate at the same time through the network. Enhancing scalability by using the current implementations, jeopardize throughput and latency.
- *Guaranteeing security:* There exist many malicious activities that target IoT networks. A practical IoT network postulates immunity to all of the plausible attacks. Current state of the art security methods mostly rely on sophisticated hash puzzles. Therefore, securing a network with resource-constrained devices incapable of performing heavy-duty computations is a challenge.
- *Protecting data privacy:* The communicated data between different entities are confidential. Therefore, preserving privacy in communication links is a necessity. Making a blockchain-based IoT network more private, usually endanger the basic paradigm of decentralization in blockchains.
- *Increasing throughput:* In an IoT network, a large number of devices are required to simultaneously communicate with each other which necessitates a network with high throughput. Increasing throughput in current implementations usually reduces scalability which is not desirable.
- *Reducing latency:* In a practical IoT network, different devices need to communicate with each other in real-time. Therefore, the network latency should be negligible. Reducing latency in current implementations usually compromises scalability, which is not acceptable.
- *Reducing computational requirements:* IoT devices are mostly resource-constrained. Current state of the art security protocols mostly rely on sophisticated cryptographic computations which is a burden for resource-constrained devices.
- *Overcoming storage limitations:* In blockchain-based networks, several nodes or sometimes all the nodes are typically required to have a copy of the ledger. However, resource-constrained IoT devices cannot store huge amounts of data.

7.2. Practical blockchain-based IoT networks

In order to successfully apply blockchain to IoT networks, practically feasible solutions are required that are scalable to large networks and at the same time yield high throughput with low latency. In addition, the implementations need to be highly secure in order to defend against possible attacks and be tamper-resistant. Besides, the implementations should be compatible with resource-constrained IoT devices that have limited computational capability and restricted storage capacity. Each of the discussed consensus methods and implementations addresses several of the mentioned issues. However, there remains the need for an implementation that addresses all the existing challenges.

A potential solution to tackle this problem is the integration of blockchain with artificial intelligence (AI) for consensus. AI-enabled blockchain is a state-of-the-art topic suggested by several researchers. Machine learning algorithms, deep learning, and reinforcement learning appear very apt to address the deficiencies of the discussed consensus methods [81]. Gupta et al. postulate the possibility of leveraging these techniques to reach an easier and faster consensus [82]. They claim that the learning algorithms can be utilized to detect non-malicious nodes as block creators. Dinh et al. have surveyed many possible approaches to realize this integration [83]. They suggest that machine learning techniques are capable of detecting malicious attacks and invoking appropriate defense mechanisms against them or isolating the compromised components. However, all of these works have been suggested as future research directions without concrete theoretical formulation or practical implementation. Successful integration of AI with a consensus method can significantly improve its security, latency, and scalability which are the main concerns in a blockchain-based IoT network.

Furthermore, several consensus methods discussed in this paper rely on choosing delegates for participation in the consensus protocol. Generally, a lot of computational, network and other resources are wasted in the delegate election process. Novel consensus methods can be devised by employing learning algorithms to choose delegates in a secure manner without waste of resources.

8. Conclusions

In this article, we reviewed the existing limitations of IoT networks and required measures to address these deficiencies. Then, we discussed the possibilities of using blockchain for addressing these limitations and securing data integrity in IoT networks. In particular, we focused on the existing consensus methods (both conventional and state-of-the-art consensus methods) and their possible applicability to resource-constrained IoT networks. We surveyed the pros and cons of different consensus methods used in blockchain implementations. By considering influential features of consensus methods on IoT networks such as latency, scalability, computational overhead, and network overhead, we categorized the consensus methods into three groups: most apt, partially apt, and not applicable to IoT networks. PoET, PBFT, and Tangle were found to be the best options for blockchain-based IoT networks because of possessing the mentioned required features for IoT networks.

However, even these methods cannot fully address shortcomings of blockchain-based IoT networks. PBFT has a high network overhead which restricts its scalability. Therefore, it is only practical for small IoT networks such as smart home. PoET's main drawback is its dependency on Intel which makes it significantly centralized compared to other consensus methods. While Tangle is claimed to address all the limitations of IoT networks, in practice, it also suffers from centralization.

Furthermore, we surveyed the well-known blockchain implementations that have already employed the discussed consensus methods. We argued how private blockchains and tangle can be a better alternative to public blockchains for IoT networks. Among the discussed blockchain implementations, Hyperledger Fabric, Sawtooth, and Iota appear more promising for IoT networks and applications since they have addressed some of the existing limitations of blockchain. Each of these implementations have addressed some of the limitations including throughput, latency, computational overhead, network overhead, scalability and privacy. However, none of them have been successful in addressing all the limitations to an acceptable degree.

We believe that in order to realize a blockchain-based IoT network in a large scale and with low latency, there ought to be either a hybrid framework which combines two or more of the already existing frameworks or a new framework with a modified consensus method. The most promising approach to improve existing consensus methods or design a new one is to employ machine learning techniques. These algorithms can play a significant role in securing consensus methods without centralization or large computational and network overheads.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Bhattacharjee, M. Salimitari, M. Chatterjee, K. Kwiat, C. Kamhoua, Preserving data integrity in IoT networks under opportunistic data manipulation, in: 15th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC), 2017, pp. 446–453.
[2] M. Salimitari, S. Bhattacharjee, M. Chatterjee, Y.P. Fallah. A prospect theoretic approach for trust management in IoT networks under manipulation attacks. ACM Trans. Sen. Netw. 16 (3) (2020), 26 pages. https://doi.org/10.1145/3392058
[3] J. Debus, Consensus Methods in Blockchain Systems, 2017, pp. 1–58.
[4] Z. Zheng, S. Xie, H.-N. Dai, H. Wang, Blockchain challenges and opportunities: a survey, Int. J. Web Grid Serv., Dec. 2017.
[5] A. Sehgal, V. Perelman, S. Kuryla, J. Schonwalder, Management of resource constrained devices in the internet of things, IEEE Commun.Mag. 50 (2012).
[6] S. Popov, The Tangle. White paper, 2016.
[7] W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D.I. Kim, A survey on consensus mechanisms and mining strategy management in blockchain networks, IEEE Access 7 (2019) 22328–22370.
[8] W. Gao, W.G. Hatcher, W. Yu, A survey of blockchain: techniques, applications, and challenges, in: 2018 27th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2018, pp. 1–11.
[9] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, M. Imran, Blockchain for cloud exchange: a survey, Comput. Electric. Eng. 81 (2020) 106526.
[10] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, C. Rong, A comprehensive survey of blockchain: from theory to IoT applications and beyond, IEEE Internet Things J. 6 (5) (2019) 8114–8154.
[11] J. Xie, H. Tang, T. Huang, F.R. Yu, R. Xie, J. Liu, Y. Liu, A survey of blockchain technology applied to smart cities: research issues and challenges, IEEE Commun. Surv. Tut. 21 (3) (2019) 2794–2830.
[12] X. Wang, X. Zha, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Survey on blockchain for internet of things, Comput. Commun. 136 (2019) 10–29.
[13] I.-C. Lin, T.-C. Liao, A survey of blockchain security issues and challenges., IJ Netw. Secur. 19 (5) (2017) 653–659.
[14] H. Zhang, K. Sakurai, Blockchain for IoT-based digital supply chain: a survey, in: International Conference on Emerging Internetworking, Data & Web Technologies, Springer, 2020, pp. 564–573.
[15] A. Miglani, N. Kumar, V. Chamola, S. Zeadally, Blockchain for internet of energy management: review, solutions, and challenges, Comput. Commun., 2020.
[16] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. White Paper
[17] O. Novo, Blockchain meets IoT: an architecture for scalable access management in IoT, IEEE Internet Things J. 5 (2) (2018) 1184–1195.
[18] Y. Xiao, N. Zhang, W. Lou, Y.T. Hou, A survey of distributed consensus protocols for blockchain networks, IEEE Commun. Surv. Tut., 2020.
[19] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, 2018, p. 30.
[20] A. Banafa, IoT and blockchain convergence: benefits and challenges, IEEE Internet Things, 2017.
[21] I. Mistry, S. Tanwar, S. Tyagi, N. Kumar, Blockchain for 5g-enabled IoT for industrial automation: a systematic review, solutions, and challenges, Mech. Syst. Signal Process. 135 (2020) 106382.
[22] T.M. Fernández-Caramés, P. Fraga-Lamas, A review on the use of blockchain for the internet of things, IEEE Access 6 (2018) 32979–33001.
[23] P. Triantafillou, N. Ntarmos, S. Nikoletseas, P. Spirakis, Nanopeer networks and p2p worlds, in: Proceedings Third International Conference on Peer–to-Peer Computing (P2P2003), IEEE, 2003, pp. 40–46.
[24] M. Ali, Z.A. Uzmi, Csn: a network protocol for serving dynamic queries in large-scale wireless sensor networks, in: Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004., IEEE, 2004, pp. 165–174.
[25] S. Krco, D. Cleary, D. Parker, P2p mobile sensor networks, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, IEEE, 2005. 324c–324c
[26] N. Kshetri, Can blockchain strengthen the internet of things? IT Prof. 19 (4) (2017) 68–72.
[27] M. Anirudh, S.A. Thileeban, D.J. Nallathambi, Use of honeypots for mitigating dos attacks targeted on IoT networks, in: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), IEEE, 2017, pp. 1–4.
[28] Q. Xu, P. Ren, H. Song, Q. Du, Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations, IEEE Access 4 (2016) 2840–2853.
[29] X. Li, H. Wang, Y. Yu, C. Qian, An IoT data communication framework for authenticity and integrity, in: 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), IEEE, 2017, pp. 159–170.
[30] T. Yu, X. Wang, A. Shami, Recursive principal component analysis-based data outlier detection and sensor data aggregation in IoT systems, IEEE Internet Things J. 4 (6) (2017) 2207–2216.
[31] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.

[32] M.A. Khan, K. Salah, IoT security: review, blockchain solutions, and open challenges, Fut. Gener. Comput. Syst. 82 (2018) 395–411.
[33] J. Lin, Z. Shen, C. Miao, Using blockchain technology to build trust in sharing Lorawan IoT, in: Proceedings of the 2nd International Conference on Crowd Science and Engineering, ACM, 2017, pp. 38–43.
[34] L. Wang, R. Ranjan, Processing distributed internet of things data in clouds, IEEE Cloud Comput. 2 (1) (2015) 76–80.
[35] M. Salimitari, M. Chatterjee, M. Yuksel, E. Pasiliao, Profit maximization for bitcoin pool mining: aprospect theoretic approach, in: Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on, IEEE, 2017, pp. 267–274.
[36] F.A. Khan, M. Asif, A. Ahmad, M. Alharbi, H. Aljuaid, Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development, Sustain. Cities Soc. (2020) 102018.
[37] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: International conference on financial cryptography and data security, Springer, 2014, pp. 436–454.
[38] Hyperledger, (https://www.hyperledger.org)Accessed: 2018-04-16.
[39] Bitshares(https://bitshares.org) Accessed: 2019-05-23.
[40] D. Larimer, Delegated Proof-of-Stake (dpos), 2014. Bitshare Whitepaper
[41] Leased Proof of Stake, (docs.wavesplatform.com/platform-features/leased-proof-of-stake-lpos.html)Accessed: 2018-04-25.
[42] Proof of Importance, (https://nem.io/technology/) Accessed: 2018-04-05.
[43] V. Buterin, V. Griffith, Casper the friendly finality gadget, 2017 arXiv:1710.09437.
[44] Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 507–527.
[45] V. Zamfir, Casper the Friendly Ghost: A Correct by Construction Blockchain Consensus Protocol, 2017.
[46] D. Mazieres, The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus, 32, Stellar Development Foundation, 2015.
[47] T.T.A. Dinh, R. Liu, M. Zhang, G. Chen, B.C. Ooi, J. Wang, Untangling blockchain: a data processing view of blockchain systems, IEEE Trans. Knowl Data Eng. 30 (7) (2018) 1366–1385.
[48] L.S. Sankar, M. Sindhu, M. Sethumadhavan, Survey of consensus protocols on blockchain applications, in: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2017, pp. 1–5.
[49] D. Schwartz, N. Youngs, A. Britto, et al., The Ripple Protocol Consensus Algorithm, 5, Ripple Labs Inc White Paper, 2014.
[50] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: architecture, consensus, and future trends, in: 2017 IEEE International Congress on Big Data (BigData Congress), IEEE, 2017, pp. 557–564.
[51] N. Chalaemwongwan, W. Kurutach, State of the art and challenges facing consensus protocols on blockchain, in: 2018 International Conference on Information Networking (ICOIN), IEEE, 2018, pp. 957–962.
[52] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea, E.B. Hamida, Consortium blockchains: overview applications and challenges, Int.J. Adv. Telecommun. 11 (2018).
[53] Tendermint, (https://tendermint.com) Accessed: 2019-03-07.
[54] J. Kwon, Tendermint: consensus without mining, Retrieved May 18(2014) 2017.
[55] L. Bach, B. Mihaljevic, M. Zagar, Comparative analysis of blockchain consensus algorithms, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2018, pp. 1545–1550.
[56] E. Syta, I. Tamas, D. Visher, D.I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, B. Ford, Keeping authorities" honest or bust" with decentralized witness cosigning, in: 2016 IEEE Symposium on Security and Privacy (SP), Ieee, 2016, pp. 526–545.
[57] E.K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford, Enhancing bitcoin security and performance with strong consistency via collective signing, in: 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 279–296.
[58] S. Micali, M. Rabin, S. Vadhan, Verifiable random functions, in: 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), IEEE, 1999, pp. 120–130.
[59] Algorand, (https://www.algorand.com) Accessed: 2019-05-07.
[60] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, Algorand: scaling byzantine agreements for cryptocurrencies, in: Proceedings of the 26th Symposium on Operating Systems Principles, ACM, 2017, pp. 51–68.
[61] M. Zamani, M. Movahedi, M. Raykova, Rapidchain: scaling blockchain via full sharding, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 931–948.
[62] T. Hanke, M. Movahedi, D. Williams, Dfinity technology overview series, consensus system, arXiv:1805.04548, 2018.
[63] Q. Zhou, H. Huang, Z. Zheng, J. Bian, Solutions to scalability of blockchain: a survey, IEEE Access, 2020.
[64] G. Yu, X. Wang, K. Yu, W. Ni, J.A. Zhang, R.P. Liu, Survey: sharding in blockchains, IEEE Access 8 (2020) 14155–14181.
[65] G. Danezis, S. Meiklejohn, Centrally banked cryptocurrencies, arXiv:1505.068952015.
[66] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 17–30.
[67] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, B. Ford, Omniledger: a secure, scale-out, decentralized ledger via sharding, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 583–598.
[68] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, C. Qijun, A review on consensus algorithm of blockchain, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2017, pp. 2567–2572.
[69] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14), 2014, pp. 305–319.
[70] IoTa Development Roadmap, (https://blog.iota.org/iota-development-roadmap-74741f37ed01) Accessed: 2018-04-14.
[71] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, A taxonomy of blockchain-based systems for architecture design, in: 2017 IEEE International Conference on Software Architecture (ICSA), IEEE, 2017, pp. 243–252.
[72] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, G. Danezis, Consensus in the age of blockchains, arXiv:1711.039362017.
[73] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.-Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends, IEEE Trans. Syst. Man Cybern. (2019).
[74] V.A. Red, Practical comparison of distributed ledger technologies for IoT, in: Disruptive Technologies in Sensors and Sensor Systems, vol. 10206, International Society for Optics and Photonics, 2017, p. 102060G.
[75] C. Cachin, et al., Architecture of the hyperledger blockchain fabric, in: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, vol. 310, 2016, p. 4.
[76] V. Buterin, et al., A Next-Generation Smart Contract and Decentralized Application Platform, vol. 3, 2014. white paper
[77] S. Roy, F.J.A. Morais, M. Salimitari, M. Chatterjee, Cache attacks on blockchain based information centric networks: an experimental evaluation, in: Proceedings of the 20th International Conference on Distributed Computing and Networking, ACM, 2019, pp. 134–142.
[78] V. Dhillon, D. Metcalf, M. Hooper, The hyperledger project, in: Blockchain Enabled Applications, Springer, 2017, pp. 139–149.
[79] M. Valenta, P. Sandner, Comparison of Ethereum, Hyperledger Fabric and Corda, Technical Report, FSBC Working Paper, 2017.
[80] C. Saraf, S. Sabadra, Blockchain platforms: a compendium, in: 2018 IEEE International Conference on Innovative Research and Development (ICIRD), IEEE, 2018, pp. 1–6.
[81] M. Salimitari, M. Joneidi, M. Chatterjee, Ai-enabled blockchain: an outlier-aware consensus protocol for blockchain-based IoT networks, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[82] S. Gupta, M. Sadoghi, Blockchain transaction processing., Encyclopedia of Big Data Technologies, 2019.
[83] T.N. Dinh, M.T. Thai, Ai and blockchain: a disruptive integration, Computer 51 (9) (2018) 48–53.