# An Overview of Blockchain and Consensus Protocols for IoT Networks

Mehrdad Salimitari and Mainak Chatterjee
*Department of Computer Science*
*University of Central Florida*
*Orlando, FL 32825*
*Email: {mehrdad, mainak}@cs.ucf.edu*

**Abstract**

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 domains are its inherent security mechanisms and immunity to different 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 apt for resource constrained systems.

In this article, we discuss and survey the various blockchain based consensus methods that are applicable to resource constrained IoT devices and networks. A typical IoT network consists of several devices which have limited computational and communications capabilities. Most often, these devices cannot perform the 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 discuss some of the alternatives to the public blockchain like private blockchain and tangle, and their potential adoption for IoT networks. Furthermore, we discuss the existing consensus methods and blockchain implementations and explore the possibility of utilizing them to realize a blockchain based IoT network. Some of the open research challenges are also put forward.

**Keywords**

Internet of Things; Public blockchain; Private blockchain; IoT networks; Consensus; Hyperledger; Distributed ledger; Tangle.

## I. 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].

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 common decision without the participation of a central controller [2]. Consensus methods typically require high computational and communications capabilities. There are some that are less demanding, however, they do not provide strict guarantees on the performance attributes [3].

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 [4]. These resource constraints make it difficult to directly apply the traditional consensus protocols to IoT networks. Newer technologies like blockchain [3] and tangle [5], 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.

This article reviews how blockchain works and argues that full-fledged implementations of blockchain (as in bitcoin and other cryptocurrencies) are impractical for resource-constrained IoT devices and networks. Though blockchain 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 methods of consensus that do not require high computational power for solving hash problems. However, being private, only valid users can access them. We compare the pros and cons of various implementations and discuss 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 II, we present blockchain and motivate how it can be applied for IoT networks with appropriate modifications. In section III, we discuss different methods of consensus. In section IV, we discuss two of the promising alternatives to blockchain. In section V, we survey the existing implementations of blockchain and discuss their pros and cons. In section VI, we put forward some of the open research challenges. Conclusions are drawn in the last section.

## II. A BRIEF OVERVIEW OF BLOCKCHAIN

A 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. Blockchains have been designed to work on an unreliable network with adversarial entities. By using sophisticated and compute-intensive secure hash algorithms, it achieves data integrity by preventing data erasure or manipulation, and invalid information from being recorded. These compute-intensive algorithms are part of a *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 methods of consensus with significantly lower computational requirement and network overhead that we will review in section III.

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

Blockchain stores all data transactions in chained blocks, i.e., a block contains several transactions linked to previous blocks by a hash pointer [7]. 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*. This architecture makes it infeasible to modify the transactions in the blockchain because in order to change a transaction, it is required to change all the blocks built on that transaction in less than 10 minutes. This modification might be feasible with

more than half of the hash rate power of the world. In other words, in order to add a block to the blockchain, 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 [6]. However, adversaries can affect blockchain operation of cryptocurrencies such as bitcoin by gaining control over 25% of computation power via an attack known as selfish mining [8]. Although this attack cannot jeopardize the immutability of bitcoin's blockchain, it can increase revenue of the adversaries.

### *A. An Illustrative Example*

Figure 1 shows an illustration of 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 0's. 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 $(\texttt{Block\#}, \texttt{Nonce}, \texttt{Data}, \texttt{Prev}) \leq \texttt{target value}$. Suppose the nonce found is 42345 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 0's. 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. 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.

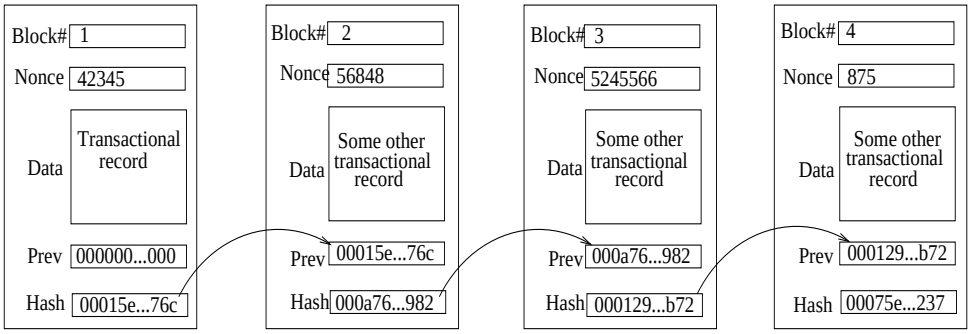



Figure 1. Chaining of blocks in blockchain

### *B. Requirements of IoT Networks*

We are witnessing the use of IoT networks in various domestic, industrial, and military applications. A common feature of these IoT networks is that they consist of 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 [9].

Most of the currently used IoT networks are based on the server-client model where all the devices are identified, authenticated and connected via cloud servers requiring enormous amount of processing capability and storage capacity. In addition, all the communications between these devices have to go through the Internet even if the devices are close to each other. Though such a model is practical for small IoT networks, it does not scale well. Furthermore, the cost of establishing large number of communication links, maintaining centralized clouds, and

networking all equipment, is significant for large-scale IoT networks. Apart from the costs, reliance on cloud servers make the architecture susceptible to single point of failure. Moreover, IoT devices must be immuned to information attacks and physical tampering. Though some of the existing methods make the IoT devices secure, they are complex and not appropriate for resource-constrained IoT devices with limited computation power [9].

### C. Blockchain for Resource Constrained IoT Networks

Blockchain establishes a peer-to-peer network which decreases the cost of installation and maintenance of centralized clouds, data centers, and networking equipment by distributing the computational and storage requirements among all the devices within the network. This communication paradigm solves the single point of failure problem. Blockchain addresses the privacy concerns for IoT networks by using cryptographic algorithms. It also solves the reliability issues in IoT networks by using tamper-resistant ledgers [9].

In spite of the built-in mechanisms that guarantee data integrity in blockchain based systems, the implementation of it in resource constrained IoT networks is challenging due to the following reasons. First, computation of the cryptographic hashes as part of the consensus method is compute-intensive and demands immense CPU cycles. Second, the communication links could become a bottleneck in delivering the transactions to others and getting their approvals. 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 consist of many devices that need to communicate with each other very quickly and at all times. This necessitates adding many blocks containing large number of transactions to the blockchain every second which requires low latency consensus methods.

Typically, resource constrained IoT systems have some flexibility in their performance requirements and ready to trade some level of data integrity for savings in computations and energy consumption. One of the ways to achieve that is to relax the proof of work in order to reduce computations requirements. Another way could be not to maintain *all* the blocks since as the number of blocks increase, larger storage is required. Rather, the last $l$ blocks could be chained and all computations could be performned on the last $l$ blocks. Though such techniques would not yield the level of data integrity usually provided by cryptocurrencies, they would nevertheless guarantee some reduced level of data protection. Other ways to empower blockchain to be used for resource-constrained IoT networks is to use methods of consensus that have significantly lower computational requirements, network overhead, and faster convergence. These methods are discussed next in section III.

## III. Consensus Methods

There are several well-established methods by which different nodes in a blockchain network can reach consensus over a new block. A blockchain based system is as secure and robust as its underlying consensus method. The most well-known consensus method is proof of work (discussed in Section II) which is used by bitcoin. Proof of work has proved to be an effective approach for cryptocurrencies over years. However, due to its high computational and bandwidth requirements, it does not seem to be practical for IoT networks. Therefore, we present other existing consensus methods and discuss the possibility of applying them to a blockchain based IoT network.

### A. Proof of Stake (PoS)

This is the most prevalent method of consensus in the blockchains used for cryptocurrencies after proof of work. This method works similar to proof of work but with a significant difference. It does not induce in a race amongst the nodes to solve the next block. Instead of a competition between the nodes to solve the next block, a node is chosen by lottery to solve the next block. The node that will mine the next block is chosen based on its proportional stake in the network which is its wealth in terms of that cryptocurrency. The chosen node will use a digital signature to prove its ownership over the stake instead of solving a complicated hash problem. As a result, this method 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 [2]. 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 stake which somehow makes the blockchain 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 for transaction fees. However, some modifications are applied to this method to deal with these problems. Although, this method has significantly eased computational requirements of proof of work, it is not yet popular for resource constrained IoT networks.

### *B. Delegated Proof of Stake (DPoS)*

This method is based on proof of stake consensus method. Contrary to PoS which is direct democratic, this method is representative democratic [3]. It means that a group of nodes chooses a node as a delegate to generate and validate blocks. It makes the transactions faster but at the cost of making the PoS blockchain more centralized. It should be noted that there exists a protocol for detecting and voting out a malicious delegate in this method of consensus [2]. A cryptocurrency called Bitshares uses this method of consensus.

### *C. Leased Proof of Stake (LPoS)*

Leased Proof of Stake 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 [10].

### *D. Proof of Importance (PoI)*

Proof of Importance 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 method of consensus considers productive network activity of nodes which is more efficient than only nodes' balances [11]. NEM is a cryptocurrency that uses PoI for consensus.

### *E. Practical Byzantine Fault Tolerance (PBFT)*

This consensus method is used for solving 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. 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 [2]. This method is well-suited for private blockchains like Hyperledger projects which are controlled by a third-party. However, it is not applicable to permissionless, public blockchains due to its low tolerance to malicious activities, hence preventing it from reaching a valid consensus.

### *F. Delegated Byzantine Fault Tolerance (dBFT)*

Delegated Byzantine Fault Tolerance follows the same rules as PBFT but it does not require participation of all the nodes for adding a block. In dBFT, some nodes are chosen as delegates of other nodes and according to some rules, they pursue the consensus protocol similar to PBFT [3]. A cryptocurrency called NEO uses this method of consensus.

### G. Proof of Capacity (PoC)

Proof of Capacity is similar to proof of work but instead of depending on computing power of miners, it relies on the hard disk space of the miners. As per PoC, miners have to store huge data sets known as plots to get the chance of mining the next block. Therefore, by saving more plots, a miner will have a higher probability to solve the next block [2]. This method is not a rational choice for IoT networks where the devices have limited storage capacity.

### H. Proof of Activity (PoA)

Proof of Activity is a hybrid method of consensus based on proof of work and proof of stake [3]. 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 [2].

### I. 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 [2]. While this approach is practical for designing cryptocurrencies, it is not appropriate for IoT applications since this method is contingent upon existence of a monetary framework and burning of coins, neither of which is inherent in an IoT network.

### J. Proof of Elapsed Time (PoET)

Proof of Elapsed Time is a consensus method proposed by Intel which works similar to proof of work but with significantly lower energy consumption. In this method, miners have to solve a hash problem similar to that of proof of work. 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) [12]. Its eased computational requirements make it IoT network friendly. The main drawback of this approach is its dependency on Intel which is in conflict with the basic philosophy of blockchain being entirely decentralized.

### K. Comparisons of Different Consensus Methods

The above mentioned consensus methods have been used in various blockchain implementations. It is worth emphasizing that blockchain implementations and their consensus methods are inseparable. The consensus method is the backbone of a blockchain implementation. Thus, most of the features and performance attributes of a blockchain implementation are contingent upon the method of consensus used. In section V-D, we present a comparison of some of the consensus methods that have already been implemented. In particular, we compare the following methods of consensus: Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), and Proof of Elapsed Time. Proof of Capacity (PoC), Proof of Activity (PoA), and Proof of Burn (PoB) do not look promising for IoT networks so they are not compared in section V-D. The other mentioned consensus methods in this section including Delegated Proof of Stake (DPoS), Leased Proof of Stake (LPoS), Proof of Importance (PoI), and Delegated Byzantine Fault Tolerance (dBFT) were modified versions of a general consensus method so they are not compared separately in section V-D.

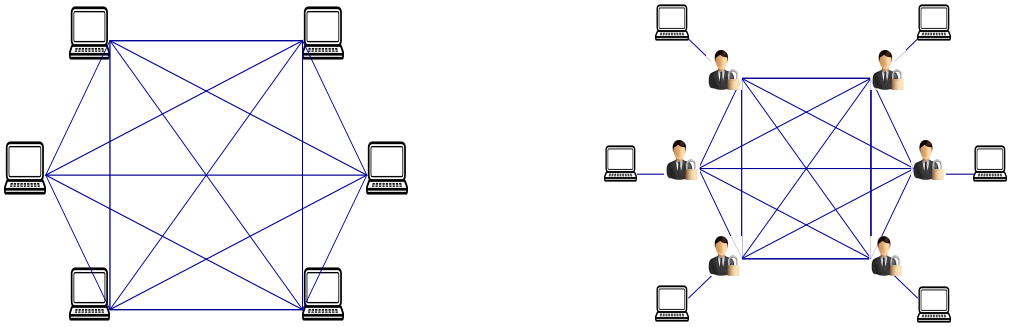
## IV. Blockchain Alternatives

There are debates over the right choice of consensus methods that are most appropriate for IoT networks. In this section, we discuss two of the most promising alternatives to the public blockchain. We discuss Private Blockchain in Section IV-A which is a feasible alternative for resource-constrained IoT devices. We discuss Tangle in Section IV-B which is similar to blockchain but uses Directed Acyclic Graph (DAG) as the basic premise for attaining consensus.

### A. Private Blockchain

There exist some blockchains designed by different companies for specific applications with restricted access to the public. The most known private blockchains are part of the Hyperledger project which is a collaboration between many well-known companies and hosted by the Linux foundation. 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 decentralization and entirely distributed. However, these blockchains have lower computational requirements and faster network response time which make them more desirable for IoT applications. 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.

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. The difference between public and private blockchains is illustrated in Fig. 2. It should be mentioned that 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. Private blockchains can follow different methods of consensus like practical Byzantine fault tolerance, proof of elapsed time, and proof of stake which have been discussed in section III.



(a) Public Blockchain (b) Private Blockchain

Figure 2. Public blockchain vs private blockchain. In private blockchain, users' access to the blockchain is restricted by an institution.

### B. Tangle

Tangle is a new technology for distributed ledgers proposed by the cryptocurrency Iota. Tangle 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. The approval of two older transactions is done by proof of work. Tangle uses Directed

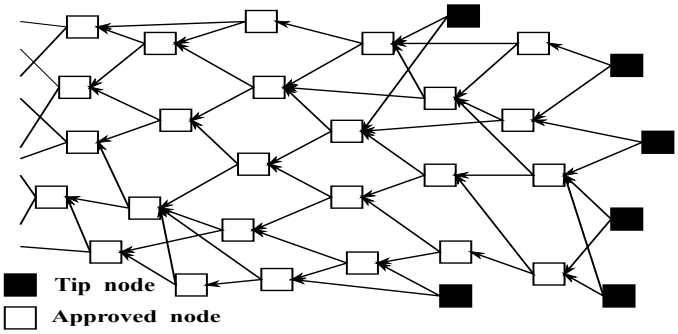



Figure 3. Tangle protocol

Acyclic Graph (DAG) in which each transaction is linked to two older transactions which are approved by it. 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. In contrast to most of the blockchain technologies using cryptographic hash codes as their method of consensus which can become obsolete with the advent of quantum computers, tangle is immune to this problem. 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 [5], it confronts a lot of implementational 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 roadmap [13]. 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.

## V. Overview of Existing Implementations

In order to achieve a blockchain based IoT network, the most appropriate framework must be designed and deployed. One option is to implement a new framework and use the preferred method of consensus 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 focus on the main difference among the various blockchain frameworks, i.e., their method of consensus. However, there are other differences including 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 [3].

The various private blockchain implementations are presented in Section V-A. 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. IBM blockchain for IoT is another private blockchain implementation. As part of the public blockchain based system, we discuss the well-known ones: ethereum and bitcoin in Section V-B We also discuss the implementations for alternative technologies for blockchain including corda and tangle based implementation of iota in Section V-C. We compare the most promising frameworks for IoT applications in Section V-D.

### A. Private Blockchains

*Hyperledger Fabric:* Hyperledger Fabric is a permissioned implementation which is widely used by enterprises. It uses a pluggable method of consensus which is defined based on specific application requirements. Its most common method of consensus is PBFT. Unlike bitcoin and ethereum which are public blockchains, this private blockchain framework attains consensus within hundreds of milliseconds [14]. Such low latency is crucial for building blockchain based IoT networks. Being permissioned, the blockchain is controlled by a specific organization which 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. 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. 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, the network overhead for this framework significantly increases with increase in the number of nodes which causes the number of messages communicated in the PBFT protocol to increase. This prevents hyperledger fabric to be used in large scale applications similar to public blockchains.

*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 [15]. However, this framework is not yet implemented in large scale and fully tested for its performance capabilities. It is also not recommended by Intel to be used for security sensitive applications due to its lack of security mechanisms.

*Hyperledger Indy:* Hyperledger Indy is a distributed ledger framework designed specifically for decentralized identities to prevent digital identity breaches on the Internet. It utilizes Zero-Knowledge Proofs (ZKP) to avoid inessential revelation of identity features. It is a permissioned blockchain but with global public access to its features. For validating new blocks, it uses an approach called Plenum in which a set of validator nodes run a modified, redundant Byzantine fault tolerant protocol [12]. There is no striking aspect of Hyperledger Indy that makes it attractive for an IoT network.

*Hyperledger Iroha:* Hyperledger Iroha is a simple implementation that focuses on mobile application development of blockchain technology [15]. It is developed in `C++` and uses a new method of consensus called Sumeragi which is a chain-based Byzantine fault tolerant consensus algorithm. In this framework, data storage and synchronization are performed off-device [12].

*Hyperledger Burrow:* Hyperledger Burrow is a permissioned blockchain that uses proof of stake for achieving consensus. It was first designed by a company called Monax. Its design is based on Ethereum Virtual Machine (EVM). It is a general-purpose smart contract machine for cross-industry applications and is not an optimal framework for a single industry [12].

*IBM Watson IoT:* Based on Hyperledger Fabric project, IBM Watson IoT is private blockchain framework for IoT networks proposed by IBM [16]. Though very appropriate for small scale IoT networks, scalability remains the main drawback. Each participant has the full solution functionality up to 10 routes (IoT to Blockchain connections), with up to 1 transaction per second per route.

### B. 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 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 which is a contract-oriented, high-level language for implementing smart contracts [17]. All the nodes are required to participate in the consensus process. It was primarily designed using 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 [14]. There are plans for migrating Ethereum to a version that uses proof of stake as the method of consensus known as Casper [3]. 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 [14]. It should be noted that some private and permissioned blockchain frameworks are also designed based on the Ethereum blockchain.

### C. Blockchain Alternatives

*Corda:* Corda is a permissioned decentralized ledger framework that uses pluggable method of consensus. 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 [17].

*Iota:* Iota is a distibuted ledger that uses Directed Acyclic Graph (DAG) instead of a blockchain. This protocol is called Tangle which was discussed in section IV-B. Its fast speed of transactions 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 [14]. 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 [5]. 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.

### D. 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 six implementations in Table I. As mentioned earlier, the most important feature of a blockchain implementation is its method of consensus which determines all the other features including computational overhead, network overhead, scalability, throughput, latency, and so on. Therefore, the comparison between blockchain implementations can be considered as a comparison between their inherit method of consensus. 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. We do not compare Indy, Iroha, Burrow, and IBM Watson since they have been primarily designed for specific purposes and are not necessarily suitable for IoT networks. Moreover, these frameworks have not yet been fully developed and implemented.

[1]These features are claimed in theory by Hyperledger Sawtooth project. However, they are not yet fully tested in a large scale implementation under different circumstances.

Table I: Comparisons of various blockchain implementations

| Implement. / Features | 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 | PoW (DAG) |
| Accessibility | Private | Private | Public | Public or private | Private | Public |
| Mode of operation | Permissioned | Permissioned or Permissionless | Permissionless | 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[1] | High[2] | High[2] | Partially[3] | High |
| Throughput | High[4] | High[1] | Very low | Low | High | High |
| Latency[5] | 100 ms[4] | Very Low[1] | 10 Minutes | 12 Seconds | Very Low[3] Not Measured | 10 ms |
| Immutability | Low | Low | High | High | High | High |
| Adversary tolerance | 33.33% Faulty Replicas | Unverified[6] | <25% Computing Power | <51% Stakes | unverified[7] | 33.33%[8] Computing Power |
| Privacy | High | High[1] | 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 |

## VI. Open Research Challenges

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. 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 through the network. Enhancing scalability using the current implementations, jeopardize throughput and latency.
- *Guaranteeing security:* There are lots of 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 rely on sophisticated computations. Therefore, securing a network with resource-constrained devices unable to perform 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.

[2]With very limited network throughput

[3]A new transaction dependent on many other transactions which node has not seen formerly will result in a significant high latency which pose some restrictions on scalability. However, generally, its latency is very low.

[4]In small scale

[5]Latency values are average values.

[6]It is still under experiment and not recommended by Intel to be used for security sensitive applications.

[7]It is claimed to be secure against adversarial attacks in Corda's white paper. However, it is not tested and implemented yet.

[8]This low tolerance for adversarial activities is solved using a special node called coordinator.

- *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 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 latency should be very low. Reducing latency in current implementations compromises scalability, which is not acceptable.
- *Reducing computational requirements:* IoT devices are mostly resource-constrained. Current state of the art security protocols rely on sophisticated cryptographic computations which is a burden for resource-constrained devices.
- *Overcoming storage limitations:* Resource-constrained IoT devices cannot store huge amounts of data which is usually a requirement for blockchain based networks. This is because, several nodes or sometimes all the nodes are typically required to have a copy of the blockchain.

In order to successfully apply blockchain to IoT networks, practically feasible solutions are required that are scalable to large networks and 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 implementations addresses several of the above mentioned issues. However, there remains the need for implementations that address all the mentioned challenges.

## VII. Conclusions

In this article, we discussed the possibilities of using blockchain for securing and assuring data integrity in IoT networks. In particular, we focused on the currently used consensus methods and their practical applicability for resource constrained IoT devices and networks. We discussed the pros and cons of current methods of consensus used in blockchain implementations. We also discussed how private blockchains and tangle can be a better alternative to public blockchains for IoT networks. Among the discussed implementations of blockchain, Hyperledger Fabric, Sawtooth, Iota, and Ethereum 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 on 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 an existing framework that have a modified method of consensus.

## References


[1] Bhattacharjee, Shameek, Mehrdad Salimitari, Mainak Chatterjee, Kevin Kwiat, and Charles Kamhoua. "Preserving Data Integrity in IoT Networks Under Opportunistic Data Manipulation." In Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl, pp. 446-453.

[2] Debus, Julian. "Consensus methods in blockchain systems." Frankfurt School of Finance & Management, Blockchain Center, Tech. Rep (2017).

[3] Zheng, Zibin, Shaoan Xie, Hong-Ning Dai, and Huaimin Wang. "Blockchain challenges and opportunities: A survey." Work Pap.2016 (2016).

[4] Sehgal, Anuj, Vladislav Perelman, Siarhei Kuryla, and Jurgen Schonwalder. "Management of resource constrained devices in the internet of things." IEEE Communications Magazine 50, no. 12 (2012).

[5] Popov, Serguei. "The tangle." cit. on (2016): 131.

[6] Salimitari, Mehrdad, Mainak Chatterjee, Murat Yuksel, and Eduardo Pasiliao. ”Profit Maximization for Bitcoin Pool Mining: A Prospect Theoretic Approach.” In Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on, pp. 267-274. IEEE, 2017.

[7] Nakamoto, Satoshi. ”Bitcoin: A peer-to-peer electronic cash system.” (2008).

[8] Eyal, Ittay, and Emin Gn Sirer. ”Majority is not enough: Bitcoin mining is vulnerable.” Communications of the ACM 61, no. 7 (2018): 95-102.

[9] Banafa, Ahmed. ”IoT and Blockchain Convergence: Benefits and Challenges.” IEEE Internet of Things (2017).

[10] ”Leased Proof of Stake”, https://docs.wavesplatform.com/platform-features/leased-proof-of-stake-lpos.html, Accessed: 2018-04-25

[11] ”Proof of Importance”, https://nem.io/technology/, Accessed: 2018-04-05

[12] ”Hyperledger”, https://www.hyperledger.org, Accessed: 2018-04-16

[13] ”Iota Development Roadmap”, https://blog.iota.org/iota-development-roadmap-74741f37ed01, note = Accessed: 2018-04-14

[14] Red, Val A. ”Practical comparison of distributed ledger technologies for IoT.” In Disruptive Technologies in Sensors and Sensor Systems, vol. 10206, p. 102060G. International Society for Optics and Photonics, 2017.

[15] Dhillon, Vikram, David Metcalf, and Max Hooper. ”The Hyperledger Project.” In Blockchain Enabled Applications, pp. 139-149. Apress, Berkeley, CA, 2017.

[16] ”IBM Watson IoT”, https://www.ibm.com/internet-of-things/spotlight/blockchain, Accessed: 2018-04-25

[17] Valenta, Martin, and Philipp Sandner. Comparison of Ethereum, Hyperledger Fabric and Corda. FSBC Working Paper, 2017.