Journal of Systems Architecture 127 (2022) 102503

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A survey and taxonomy of consensus protocols for blockchains

Arshdeep Singh [a], Gulshan Kumar [a,b,*], Rahul Saha [a,b], Mauro Conti [b], Mamoun Alazab [c], Reji Thomas [d]

[a] School of Computer Science and Engineering, Lovely Professional University, Punjab, India
[b] Department of Mathematics, University of Padua, 35131 Padua, Italy
[c] Charles Darwin University, Casuarina, NT, Australia
[d] School of Chemical ENgineering and Physical Sciences and Division of Research and Development, Lovely Professional University, India

ARTICLE INFO

Keywords:
Consensus
Blockchain
Bitcoin
Protocol
Decentralization

ABSTRACT

Blockchain is an emerging decentralized and distributed technology. Along with the beneficial features of decentralization, transparency, and security the consensus algorithms of blockchains form key building blocks for this technology. Consensus protocol/algorithm helps to provide a decentralized decision making process. An efficient consensus algorithm is inclusive that engages all the participants to make their decision based on the conflicts of the blockchain networks. These consensus decisions lead to better quality outcomes of the blockchains and help to obtain the finality. Rigorous research is in process to upgrade or optimize the existing consensus protocols. The optimized or enhanced consensus protocols objectify to be suitable for Internet-of-Thing (IoT) as the current versions of the protocols are not suitable for the resource-constrained environments due their complexity, hard configurations, mining techniques, high resource consumption, and explicit security loophole.

In this paper, we present a survey of consensus protocols with a purpose to identify and discuss the existence of various consensus protocols available in literature. We emphasize on the genesis of the consensus protocols, particularly for Proof-of-X, byzantine fault tolerance, Paxos, and RAFT; we also include Directed Acyclic Graph (DAG) orientation of some contemporary algorithms. We discuss the variants of these genesis protocols. Our survey analyzes the advantages, disadvantages, and their applicability in IoTs. We enlist the categorical use of consensus algorithms in blockchains and other applications. Finally, we present several research trends and open issues emphasizing for consensus protocols emphasizing on IoTs. Compared to the other surveys in the field, our present survey objectifies to provide a more thorough summary of the most relevant protocols and application issues; this survey helps the researchers and the application developers to obtain an insight on the current status of the consensus protocols' suitability to deliver the desired functionalities in IoTs. The notified disadvantages of each of the protocol provide future scope for the industries and academia. To the best of our knowledge, such a comprehensive and summarized survey of consensus protocols including DAG-based protocols is unavailable in the literature and thus, our contribution claims are significant.

1. Introduction

The collaboration of computing and telecommunication technologies has resulted in the evolution of computer networks. A single server or computer resource provides a drawback to suffice the need of many individuals or consumers. To solve this problem, use of multiple CPU's comes into existence with a decision making technique *Leader replica schema* or *Peer-to-peer schema* [1]. Based on the topological characteristics of a computer network, networks can be segregated into following types: *Centralized*, *Decentralized*, and *Distributed*. The need of scalability, fault tolerance, availability, replication, and reliability of data has shown the shift of centralized paradigm to the decentralized and distributed ones [2]. However, security is always considered as a pivot concern in each of the paradigms. Various discoveries in the cryptography and improvements in secure communication lead to the evolution towards the blockchain technology. Public key cryptography, cryptographic puzzles, and Rivest–Shamir–Adleman (RSA) cryptosystem are the important cryptographic discoveries which also become security core of many technology frameworks such as blockchain. In 2002, a paper by Adam Bach proposed HashCash for protecting against Denial of service attacks and it leads to Proof of Work (PoW) algorithm

[*] Corresponding author at: School of Computer Science and Engineering, Lovely Professional University, Punjab, India.
E-mail address: gulshan3971@gmail.com (G. Kumar).

https://doi.org/10.1016/j.sysarc.2022.102503
Received 26 August 2021; Received in revised form 10 March 2022; Accepted 30 March 2022
Available online 29 April 2022

used in Bitcoin. In 2008, Satoshi Nakamoto releases a white paper on Bitcoin. It proposes for online peer-to-peer transaction directly between the sender and the receiver. It is considered as the foundation for the blockchain technology [3]. From this point, the blockchain technology has evolved through various phases. This is the first phase of blockchain referred to as *Blockchain 1.0* where the concept of bitcoin and cryptocurrency gets popularized. With the increased need, the concepts of Blockchain 1.0 evolve to the next step towards *Blockchain 2.0*. It introduces smart contracts and Ethereum. The third phase is *Blockchain 3.0* in which diversified decentralized applications emerge. Apart from the cryptocurrencies, blockchain extends to other domains such as supply chain, finance, healthcare, real estate, asset management, and insurance industry. This integration of blockchain and various domains is called as *Blockchain 4.0* [4]. Various advantageous features such as decentralization, distributed, transparency, immutability, integrity, and security have provided the acceptability of blockchain in different domains [5–10].

In a blockchain, the function of validation and confirmation of the transactions is performed by the collective decision of all the members in the network ensuring safety as well as reliability and non-tampered. This is known as consensus in blockchain. It works as decentralized as well as distributed [11]. Consensus makes the blockchain more trustable for making decisions. Moreover, the removal of centralized authority from the generic transaction framework (for example, centralized banking) for decision makes the consensus less vulnerable and less maliciously influenced. Various consensus protocols are available till date. In this paper, we focus on such consensus protocols and their features. In some consensus approaches, replicated log is used to ensure that state machines (or nodes in the network) execute the same commands in the same order [12,13]. If the log is similar, the state machines execute the same command and produce the same results. This continues to perform well as long as majority of the machines are up [14]. The choice for underlying consensus algorithm depends on the blockchain type and its use [15–17].

1.1. Motivation and contribution

Consensus protocols are integral parts for Distributed Ledger Technology (DLT) such as blockchain. Depending on various parameters of transactions, nodes, and network environment various consensus developments exist. Some of the consensus protocols are having enough potential to be adopted by various applications. As we are in the age of Internet-of-Things (IoTs) and progressing rigorously with ubiquitous and pervasive computing, the use of small and integrated devices is more practical [18–20]. As a result, the available consensus protocols are not suitable to support IoT functionalities due to their tendency of high resource consumption, explicit hardware use, mining process, etc. To identify the pros and cons of the existing consensus protocols, we have surveyed the protocols available in the literature. This survey becomes a one-stop solution for the blockchain developers, more precisely consensus developers, to further extend the efficiency or to use a consensus protocol for IoT applications. The major contributions of our survey are as follows.

- We provide a concise understanding of the blockchain basics and then survey the existing consensus protocols. We consider the readers' understandability for this survey so that new readers can obtain a comprehensive pathway for consensus algorithms.
- We show an extensive taxonomy of the consensus protocols. The taxonomy has five categories: Proof-of-X, Paxos, RAFT, Practical Byzantine Fault Tolerant, and DAG. The categorical understanding of the consensus protocols is also a benefit of this survey.
- Each category of the taxonomy discusses advantages, disadvantages, and the applicability of a consensus algorithm in IoT paradigm. This discussion helps and motivates the research community to follow some open research problems. The new research dimensions lead to a futuristic pathway of consensus protocol enrichment for IoTs.

1.2. Organization

We organize the remaining part of the paper as follows. Section 2 discusses some fundamental aspects of blockchain. This includes core components, blockchain characteristics, and blockchain stack. Section 3 surveys the consensus protocols which are based on proof of some parameters. We call it *Proof-of-X*. We have surveyed 30 protocols in this category. Section 4 surveys 11 consensus protocols which are based on Paxos. Section 5 discusses about RAFT consensus. Section 6 shows the pros and cons of nine byzantine fault tolerance based consensus protocols. Section 7 reviews the consensus protocols developed for DAG-based decentralization. Section 8 analyzes the consensus protocols based on applications, blockchain types, and consensus types. It also compares the existing consensus surveys based on their attributes. Section 9 notifies some open research problems for the readers. Section 10 concludes the paper. For the ease of tracking the sections and their respective contents, we have provide a chart of paper organization in Fig. 1.

2. Blockchain fundamentals

In this section, we discuss about the fundamental understanding of blockchains. In particular, Section 2.1 discusses the core components of blockchain technology, Section 2.2 summarizes the characteristics of the blockchain technology, and Section 2.3 provides a brief overview of the blockchain stack.

2.1. Core components of blockchain

Blockchain is tamper evident and resistant digital ledger implemented in a distributed fashion without a central authority. Blockchain is comprised of the major components discussed below.

Blocks: When a publishing node publishes a block, transactions are added to the blockchain. A block contains a block header and block data. *Block header* contains metadata that may include details like block number (block height), hash value of previous block header, timestamp, nonce value, and size of block. *Block data* contains list of transactions which have been included in the block after validation and authentication. The *nonce* has its importance. In PoW, the validation of a block depends on this nonce value. The miners (nodes who calculate the hash value) must get a hash value which is equal to or less than a certain value. To achieve this goal, the miners keep changing the nonce values till the required hash is produced. Once produced, this hash is broadcast across the network where the other miners confirm it and add this block to the blockchain [21]. Thus, different message digest (hash) can be produced for the same data value by changing the nonce value [17]. The block header may also contain details like Merkle Tree Root (MTR) hash and block version. MTR hash represents the hash value for all the transactions stored within that block [21]. The current block is related to the previous block as the hash value of the previous block header is stored in the block header of the current block. We call this as *chaining* of blocks. With time, this chain of blocks increases. The first block with no parent block is called as the *Genesis Block*. Many miners in the network can be successful in producing the valid hash but only one block can be added to the chain at a time. Only the longest chain from the genesis block is taken as the current status of blockchain [22]. Fig. 2 shows a logical interpretation of the blocks and chaining.

Cryptographic Hash Functions: On passing the input data (file, text, etc.) of a variable size through a cryptographic hash function, it generates a unique output of fixed size called as hash (or message digest). This process is called as hashing [17]. It should have an effective avalanche effect [23]. The most commonly used hashing technique is SHA-256 (Secure Hash Algorithm). In bitcoin, hash function is used to point to the next block containing transaction. These hash functions

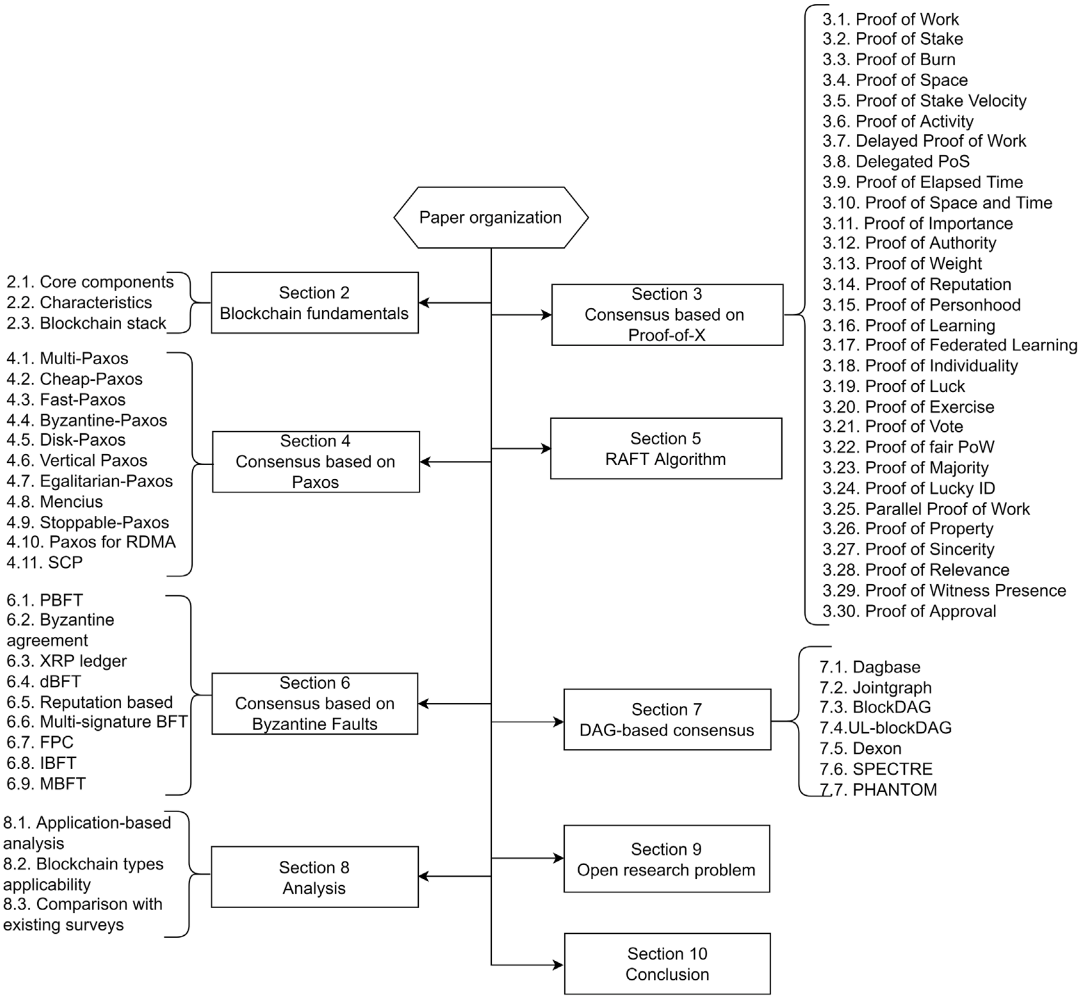


Fig. 1. Paper organization and taxonomy of blockchain consensus protocols.

enable to create a chain of blocks as shown in Fig. 2. Any modification in the block or chain also changes the hash value [24].

Merkle Tree: It is composed of hashes of different blocks of transactions. Every transaction in a blockchain has a hash that helps to connect with the other blocks of transactions and thus, the chaining occurs. These hashes are organized in a tree-like structure such that each hash is linked to its parent following a parent–child relation. The benefit of using the hash is that any change in even a single transaction (addition or deletion or modification) changes the hashes of all the transactions [25].

Asymmetric-Key Cryptography and Digital Signature: Asymmetric key cryptography uses a private–public key pair for cryptographic protection of data. The keys are such that private key cannot be derived from knowledge of its public key or vice versa. Private key (possessed by user only) is used to sign a transaction to form digital signature and this can be verified by using the public key (this key is made public for anyone to use) associated with the user. This enables members to verify that user who performs transaction has the private key [17]. The role of cryptographic techniques is to ensure that user identity is authentic and the integrity of the transactions is maintained [25].

Private Key Storage (Wallet): A wallet stores private and public keys of a user [17]. A loss of private key results in the loss of digital assets associated with that key. Since a blockchain data is immutable, any transaction or activity done with private key will be permanent and cannot be undo [25].

Node: A node may be defined as an individual system within a blockchain network. When a transaction is made by the user to the blockchain network, a software in the form of an application, digital wallet, web service, etc., is utilized to send this transaction to the nodes in the network which can then be propagated across the entire network [17]. Another method of explaining a node is that, for PoW-based blockchain, the node is one that uses CPU and resources to solve the computational problem for finding a new block [11]. We show the node categorization [26] in Fig. 3.

Distributed Ledger: With cryptocurrencies, the transactions are visible to everyone and are stored in distributed storage. This is called as *Distributed Ledger*. Each member has a copy of the global data or this distributed ledger. The users across the blockchain network maintain consensus on the version of the record of transaction along with maintaining a copy of the ledger [25].

Consensus Algorithm: Blockchain technology eliminates the role of a third party for verification of the state of system. It gives this right to each member of the network to validate and verify the integrity of system. As a result, there must be a consensus among different members of the blockchain network. This consensus is essential for adding a block to the blockchain as well as maintaining the state of the system. The use of consensus algorithm varies with the type of blockchain network (i.e., private or public or consortium). These consensus algorithms must solve the three problems associated with consensus: first, an agreement on choosing a correct block by different correct process or nodes; second, validating that the chosen block is one that is proposed by one of the process or nodes within the network, and third, all the correct process or nodes reach to a decision within

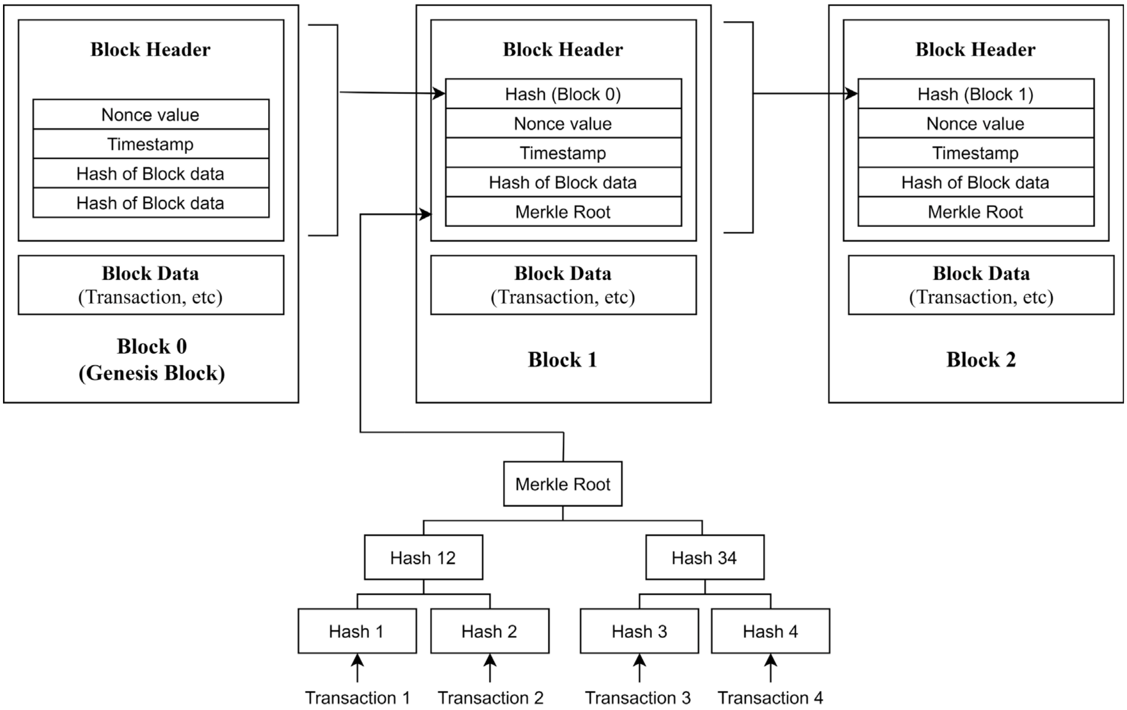


Fig. 2. Diagram representing blockchain structure.

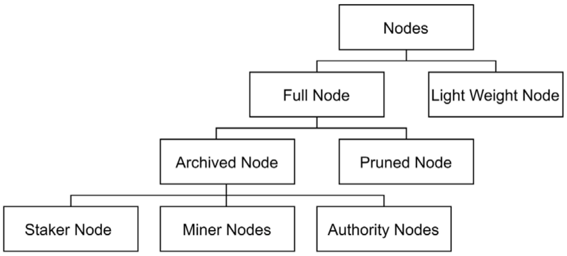


Fig. 3. Classification of nodes in a blockchain network.

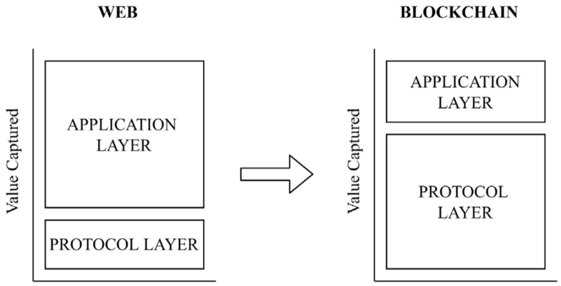


Fig. 4. Re-distribution between the application layer and protocol layer from web to blockchain.

a certain time [27]. A consensus algorithm must be designed for: *Correctness*, *Conciseness*, *Efficiency*, and *Understandability* [13].

Incentives: In some of the blockchain technology, incentives are given to the validators or verifiers as it costs them energy and money for performing the required task. In PoW, it is an effective way to retain the honest miners in the network [11].

Peer to Peer Network: It is a group of interconnected computers or devices (nodes) connected through wired or wireless connection. These nodes take part in transfer of data without the role of central server. This makes it successful against single point failure attacks [28].

Memory pools (mempools): It is a collection of pending transactions; such pool is associated with each node or member of the network; a node chooses a transaction from this pool to add to the block or broadcast across the network. Once a transaction is added to the block then it is removed from the mempool or memory pool of the node [29].

2.2. Characteristics of blockchain technology

The key characteristics of blockchain technology are as follows.

Decentralization: As already discussed in the previous section, the blockchain technology utilizes distributed system that does not require a central authority [21].

Anonymity: The members of blockchain network can interact with one another without revealing their real identity. Instead they use a generated address (public key) to communicate [21].

Auditability: Since the transactions are interconnected, current transaction must refer to the previous transaction. This leads to a chain of transactions and thus, a transaction can be verified and tracked at any point of time [21].

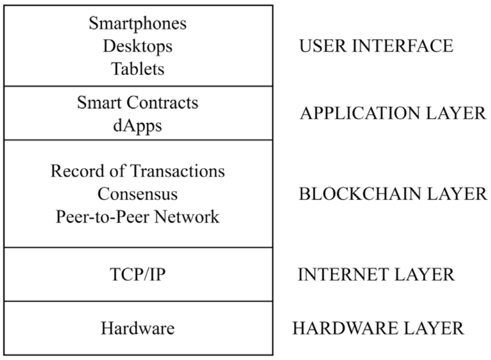
Security: The blockchain technology is secured using cryptographical techniques (like private and public keys) which ensure that data within the ledger cannot be tampered and stays attestable [17]. Since it is decentralized, blockchain protects the network against single point of failure [29].

Increased capacity: Blockchain has definitely increased the capacity of network by better involvement of the computers or devices in that network to perform a function [29].

Transparency: Everyone in the network should be able to confirm that transaction has occurred in the network [30]. The public key of user linked with transaction is visible to all. Thus, all transactions are transparent.

Reliability: A copy of transactions is stored in the distributed ledger of each node in the network. The copies of the ledger must be consistent across the network. Thus, information can be retained even if one or few members undergo failure [29].

Immutability: Once a block has been added to the blockchain, it cannot be undone [31]. Thus, a transaction once added and recorded cannot be altered [29].



Smartphones Desktops Tablets	USER INTERFACE
Smart Contracts dApps	APPLICATION LAYER
Record of Transactions Consensus Peer-to-Peer Network	BLOCKCHAIN LAYER
TCP/IP	INTERNET LAYER
Hardware	HARDWARE LAYER

Fig. 5. Blockchain stack of Ethereum and similar applications.

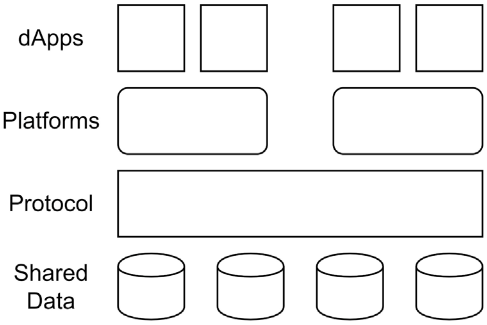


Fig. 6. Different components of the blockchain technology stack.

2.3. Blockchain stack

In blockchain stack, there is a change in distribution of value from internet stack to blockchain stack due to which it contains fat protocol layer and thin application layer [25]. This distribution is represented in Fig. 4. Speculative token attachment results in cycle that promotes application building and adds to the value associated with the protocols [32].

An explanation of blockchain chain stack [33] discusses three main layers of blockchain stack: *Internet Layer*, *Blockchain Layer* and *Application Layer*. In the internet layer, different devices including computers, tablets, phones, etc., are interconnected with one another and communication between them takes place using the TCP/IP protocol. In the blockchain layer, there is involvement of blockchain fundamentals such as consensus rules, peer to peer network, and distributed ledger of the transaction [33]. Addition of this layer results in reduction of the functions that was earlier reserved for application layer like governance, settlement, etc. [25]. The applications are built over this layer and forms the application layer [33]. Decentralized applications (dApps) along with smart contracts are built on top of blockchain layer [25]. A user interacts with these developed applications [33]. This interaction between the user and protocol take place using the products like dApps built by developers [25]. Fig. 5 shows layer wise blockchain stack. We present another view of the components [25] for a blockchain stack in Fig. 6. It shows the blockchain technology stack consists of four components: shared data, protocol, platform, and products. It stores the transactions in distributed ledger in hashed format and this shared data forms the first component. Protocols provide infrastructure for implementing rules for reaching consensus, providing incentives, and facilitating the participation process. Platforms are used as base for developing products over it that can be used by the users. Ethereum and NEO are the examples of such platform. The products include dApps along with smart contracts [25].

3. Consensus based on Proof-of-X

In this section, we discuss about the consensus algorithms which use proof of something. This “something” includes various criteria as discussed in the following subsections.

3.1. Proof of Work (PoW)

In PoW, the sender (prover) has to prove that it has done some amount of computational work in a certain interval of time [34]. The idea for this consensus is derived from the work in [35]. PoW has been proposed to be used in peer-to-peer version of electronic cash system (Bitcoin), where the online payments occur between two parties directly without any intermediary in between. It is believed to solve double spending problem that was caused due to reversible nature of online transactions [36].

Bitcoin can be defined as online currency based on accounting entries [37,38]. Transactions in blockchain occur at time intervals and a specific confirmed transaction is included in a blockchain. Each block contains a set of transactions and the addition of this block to blockchain is time and energy consuming. Each block is indexed using its hash value and the hash value of previous block is contained in every new block. This addition of block is called as the mining process and the nodes performing this operation are called as miners [39]. The miner must choose a random nonce value and calculate the hash value. If this hash value is less than a certain pre-defined target value, then the block gets added to the blockchain. This is also confirmed by the other miners in the network. In bitcoin, SHA-256 hash function is used [40]. The difficulty for calculating a valid hash is maintained by setting the target T value for every 2016 block [40]. Fig. 7 represents the process of changing the nonce value by the miner to get the desired hash value in order to mine the next block. At times two miners can add block at the same time. When two or more miner verify the same block at same time, the block that is the most synchronized in the network gets the finalization of agreement by all the nodes in that network [41].

Analysis of PoW shows that it takes 10 min to generate a block and it takes 1 h (60 min) to confirm one block; the time one block takes for confirmation, 6 blocks are generated [42]. Applications of PoW includes: Bitcoin, Ethereum, etc. In addition to its digital form of currency, Ethereum also provides platform for building applications over it.

3.2. Proof of Stake (PoS)

A proposal for PoS uses the number of bitcoins owned by a user is preferred over sharing computing resources for voting proof in accepted transaction history [43]. The miner (here we call it as validator) must hold a certain number of bitcoins and this amount determines the probability of a node getting selected as a validator for the next block. If there is a detection of malicious activity being done by a node, then the locked-up stakes get slashed or withheld. This may create a problem as the node with the largest stake has a higher chance to get selected as a validator node. Thus, unlike POW, PoS can punish the participants for performing a malicious operation and are also held accountable [44]. PoS uses digital signature to prove the ownership of coins [45]. Ouroboros is a member of PoS family. The focus of this consensus is to maintain persistence and liveliness [46]. The key idea of Ouroboros is that there are epochs containing a sequence of slots with their slot leaders who take part in block creation [47]. Cardano uses this consensus and is used as an infrastructure for maintaining the Ada cryptocurrency on Cardano [48].

One problem associated with PoS is that probability of block being selected as validator is directly proportional to the share of bitcoins owned by the validator node. To avoid such problem various unique additions have been proposed such as: *Randomized Block Selection* and *Coin Age method* [49]. Some examples of implementation of PoS include Peercoin (PPCoin), Nxt, and BlackCoin [50–52].

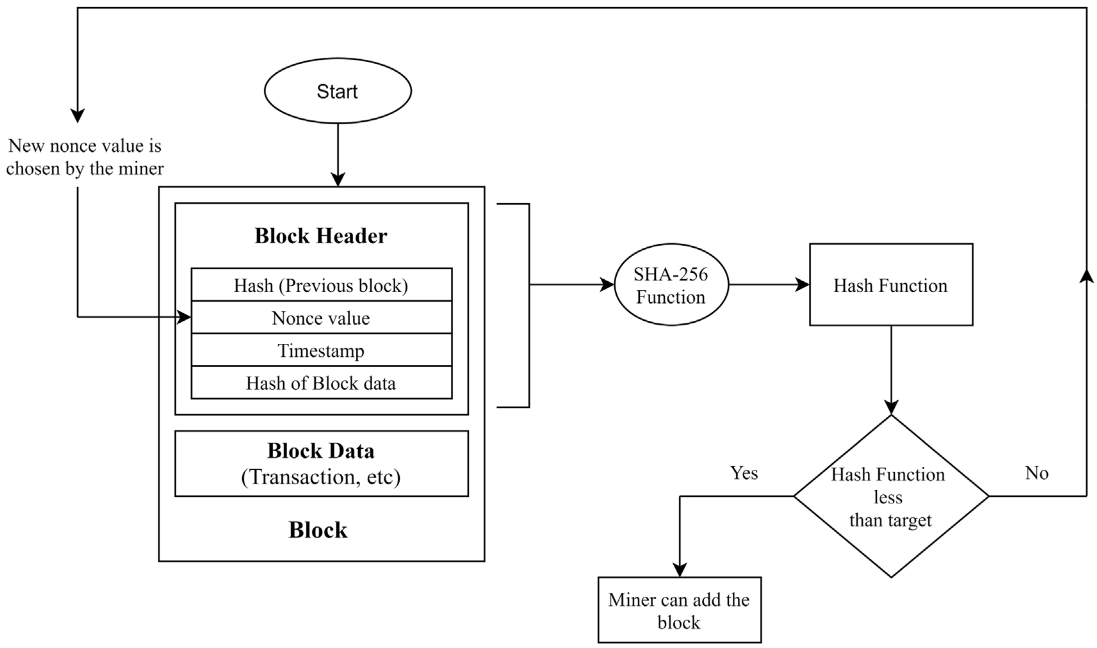


Fig. 7. Representation of the mining process.

3.3. Proof of Burn (PoB)

The concept of PoB is proposed by Iain Stewart. It is a method for distributed consensus used as an alternative to PoW and PoS [53]. In this consensus, the coins are burnt and burn hashes are generated via method exclusive to burn transactions [54]. The method does not require to use powerful computational resources and hardware. Only one calculation is required as burn hash. A burn hash is calculated as the product of multiplier and the internal hash; the value of multiplier varies with each burn transaction. This multiplier results in decay of the burnt coin. The burning of coins indicates commitment of user to the network enabling it to gain rights to mine and validate transactions. Burning can be done using bitcoins and other native coins as well. By burning, it means sending coins to a burn address (transaction to a public verifiable address where it is non-disposable) from where they cannot be traced back. Alike PoS, PoB uses block validators to invest coins for consensus mechanism. The only difference is that in PoS if the node decides to leave the network, it can take back its coins and spend it somewhere else. In PoB, there is no such relief as once the coins have been burnt (sent to an address), the coins are gone forever [45]. Some cryptocurrencies which use PoB are: *Counterparty* [55] and *Slimcoin* [56,57].

3.4. Proof of Space/Proof of Capacity (PoSC)

PoSC is an alternative concept for PoW [58]. Unlike PoW, PoSC uses space in the form of hardware space which is available at low price. The benefit with this mechanism is that people will have a certain amount of free disk space at any time available [59]. An illustration of this mechanism is available in [58]. Consider a situation between a sender and the verifier. Sender is assumed to be an organization offering free mail service while the verifier role is taken by users. The goal is to prevent registration of fake email address using this mechanism. To do this, the verifier sends a pseudorandom file (say 100 GB) to the prover during initialization phase. At times, the prover is asked to send some bits of this pseudorandom file at random positions such that verifier stores this file. The prover must show that it has the access to the memory in the form of read or write operations [60]. In Proof of Capacity, the miner must compute only one time (plotting) and the results of this work are stored on hard disk in cache form. The Hard-disk Drive (HDD) reads through the cache file for a few seconds for each block and rest of the time it remains idle [61]. A security loophole is that prover has to send only bits of the stored file. If the prover deletes the file sent by the verifier after initialization and stores only the bit or short file that needs for communication during initialization phase, this is considered as a cheating done by the sender [58].

One possible application is mentioned in the use of forensic analysis or device attestation for remote confirmation of success of wipe command for a device [60]. The possible benefits in comparison to PoW include unused storage gets used and the use of costly hardware like ASICs can be avoided. This new concept is called farming that leverages the use of existing vacant hard disk space distributed across various nodes. It is different from mining that involves wastage of electricity and tendency for centralization by one hardware. Farming does not require much bandwidth or CPU [59]. The Proof of Capacity consensus algorithm is being implemented by the Burst Blockchain [61]. It is decentralized, eco-friendly, energy efficient, and scalable [62].

3.5. Proof of Stake Velocity (PoSV)

PoSV is introduced as an alternative to PoW and PoS. Reddcoin (RDD) launched in 2014 [63,64] uses PoSV. The design for PoSV is obtained by the combination of ownership (stake) and activity (velocity). The main reason behind PoSV are the limitations made by PoS [64]. In 2019, PoSV transitioned into PoSV-v2 that aims to provide better network growth and stability [65].

3.6. Proof of Activity (PoA)

The proposal for PoA is made by Charlie, in 2012 [66]. The term “activity” denotes that reward to be given to only those active stake-holder nodes which maintain full online node. PoA is a consensus protocol designed to have a decentralized cryptocurrency in which the entities performing computational tasks gain decision making power based on PoW while the entities that hold stake gain the decision-making power based on PoS. Note that, here is that the miner tries to generate a nearly empty block header (containing some header information only). Once this block is chosen, then its block header is

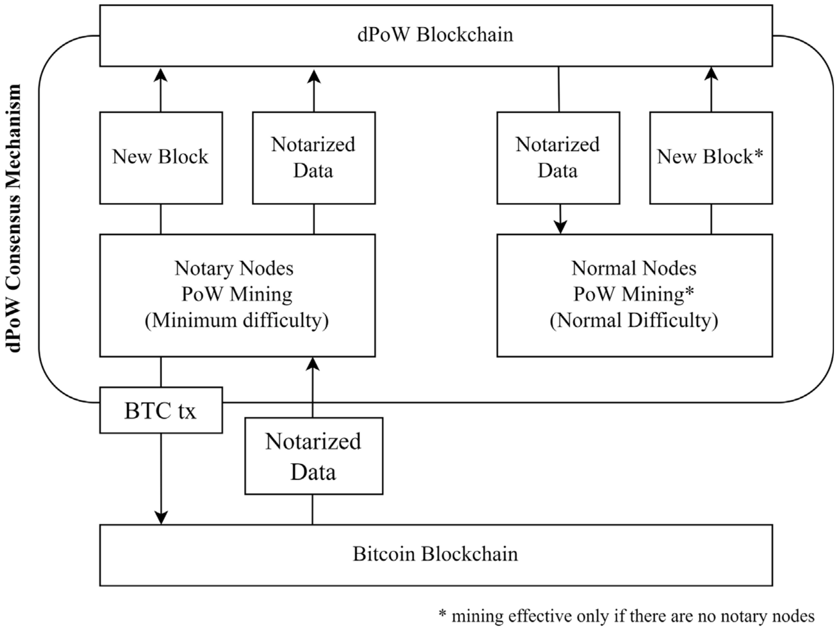


Fig. 8. Representation of the mining process in dPoW.

broadcast across the network and the next phase of signing and validation begins. The probability of validating or signing the new block depends on the percentage of stake owned based on the PoS [67]. For an attack to be successful on this combination of algorithm, the attacker has to own more than 51% of network mining power as well as more than 51% of coins must be staked [68]. The fees from the transaction is shared among the miners and the N lucky stakeholders [69].

One drawback of PoA is that the use of PoW still demands a certain amount of computational power. The method is resistant against DoS attacks [67]. Decreed is an autonomous cryptocurrency based on PoA [70]. Decreed provides hybrid PoW–PoS consensus mechanism in which every participant can make decision regarding direction of the currency making. Thus, it solves the problem of increased centralization as seen with Bitcoin or other cryptocurrencies. It aims to reduce hard forks [71].

3.7. Delayed Proof of Work (DPoW)

DPoW is not directly considered as a consensus algorithm; however, it is a security mechanism applied in addition to PoW consensus that makes blockchain more secure and resistant to 51% attack. This modified version of PoW is designed by the Komodo Project under Komodo Cryptocurrency project in late 2016 [72]. Thus, at core it can be either a PoS or PoW. The dPoW blockchain is secured by the Bitcoin's hash rate [73]. One thing must be made clear that dPoW does not participate in mining of the blockchain being secured by it. It only acts as an additional layer of security for a UTXO-based blockchain [74]. Fig. 8 shows a logical interpretation of DPoW. There are two set of nodes, notary nodes and normal nodes, with separate functions. Based on the node classification, the difficulty level also changes to smoothen the consensus process. This allows the mechanism to work even if the notary nodes undergo any sort of failure but, under such situation, the additional security linked with the blockchain is lost.

In DPoW, a block hash from a block in Komodo (KMD) chain is written onto the Bitcoin blockchain every ten minutes. It is basically a snapshot of its own blockchain which gets converted into block on bitcoin network. This step is called as notarization [75]. In this step, the notary nodes, which are elected by Komodo's community, write the block hash from dPoW protected blockchain upon the Komodo Ledger and this is completed by executing a bitcoin transaction. The notary nodes choose time of ten minutes old to ensure validity of block by the network. This increases the security as the attacker will be required to break the Bitcoin network first before breaking into the KMD chain [76]. After each notarization step, the consensus rules (example: Longest chain rule in PoW) of the blockchain are reset by the DPoW.

3.8. Delegated Proof of Stake (DPoS)

DPoS is designed by Larimer and implemented for the first time in the BitShares project [76]. The DPoS system is maintained by the coinholder that votes for delegates (also called witnesses or block producers). These delegates are responsible for validating new blocks and get reward. A fixed number of delegates are voted by stake holders. Each stakeholder gets votes proportional to the number of coins they own. Each delegate is scheduled for a certain interval of time. In case of failure in validating blocks or any malicious activity, the delegate is replaced by another delegate. Some systems implementing DPoS may require the delegates to store certain amount of coins as stake to show their commitment towards the network. In comparison to PoW, DPOS has higher performance and is more energy efficient [77].

A certain number of witnesses are elected by the stakeholder to generate blocks. Once 50% of voting stakeholder believe that there is sufficient decentralization, the top N witnesses approved are selected. The witness is paid for producing each block and this payment rate is set by stakeholder via the elected delegates. Failure to produce a block is recorded and may result in voting out in future from the witness list [78]. DPoS is used in Bitshares, Lisk, and Steem [79–81].

3.9. Proof of Elapsed Time (PoET)

The concept of PoET is proposed by Intel. The idea of this consensus is leveraged on the Hyperledger Sawtooth Lake based on SGX by Intel [82]. Sawtooth is a permissioned blockchain network technology. A trusted execution environment is used by the system to generate random wait times; this makes PoET energy-efficient than PoW [83]. It supports scalability to a larger network [84]. The validators in the blockchain request for wait time (the time it waits before generating a block) from a trusted function and one with the least value of wait time is elected as a leader node [84]. Two functions are used here: CreateTimer and CheckTimer. The former creates a timer for transaction block while the latter verifies that validator has waited for a certain time before becoming the leader [85]. The waiting time needs to follow a probability distribution to be determined by the scheme. Multiple blocks can be generated by such a waiting time until it is updated. To make sure that user has to wait for some time, each user has to generate proof of waiting along with the block to be submitted. In addition to this, statistical tools can be employed to confirm the probability distribution of the waiting time of the users [82]. Here, the minimum wait is fixed while local average wait is adjusted based on the number of active nodes. There is uniform distribution of r within range [15,82]. Increasing the waiting time on presence of more active nodes results in potential collisions. Cost is reduced by use of random waiting time multiple times. The drawback associated with this protocol is that the security of PoET based blockchain depends on the underlying computing hardware which may lack reliability and can be prone to attacks [82]. Hyperledger Sawtooth act as a blockchain platform over which distributed ledger applications and networks can be built [83,84]. It uses PoET consensus for leader election lottery system.

3.10. Proof of Space and Time (PoST)

In 2019, a green paper revealed by Chia gives an outline of construction of PoST [86]. In Chia network, these blocks are farmed and not mined. Users on network generate hashes which are stored on the disk space. On comparing with the other hashes in the network, the closest hash is chosen. With PoST, it provides an eco-friendly approach in comparison to PoW in terms of energy consumption. Spacemint blockchain, based on PoSpace, assigns quality to each PoSpace and this quality is determined by the hash of the proof. A hash with a low value is considered to be of good quality. For the block to be finalized, it is augmented with output of a Verifiable Delay Function (VDF). This concept is used in Chia where this VDF is a time parameter specifying the steps required for computation of output. Thus, it takes PoSpace, VDF(time), and a unique digital signature to form the Chia blockchain [87,88].

3.11. Proof of Importance (PoI)

PoW and PoS have a drawback that the rich user has a higher probability to sign the next block; this further increases the future chances of selection as a signer. PoI assigns user with a trust based on which they are being rewarded. It considers the amount of transactions done to others and to whom [89]. It is used by NEM cryptocurrency that is aspired to become everyday transaction medium [90]. Transactions and blocks are associated with timestamps in NEM. Creation of new blocks is called as harvesting and those who perform this operation are harvesters. Only an account with more than 10,000 XEM is eligible to harvest and all such accounts have non-zero importance score. Here, XEM is the cryptocurrency used in NEM [91]. NEM Infrastructure Server (NIS) nodes form the backbone of the NEM network in which transaction occurs only when a person controlling an account signs that transaction. The signing key is to kept in person's computer only and signing must take place in his computer only [92]. NEM can be described as a blockchain for smart asset management with high performance. Its usage includes digital assets (token, contract, etc.) transfer from one private network to another private network via a public blockchain, financial asset management, supply chain management, retail store maintenance, etc. Its APIs compatibility allows linkage between global applications and NEM platform. It also acts as a link between private and public blockchains [93].

A rough idea of the factors which contribute to the importance factor in PoI include stake owned by a user, quality transactions with others, and size and frequency of transactions. We show this logical interpretation in Fig. 9 following the work of [94]. PoI algorithm then puts together the values of the mentioned parameters and computes a rating for each node. According to the rating, the node has a certain probability of random selection for the next round of block harvesting. If the selected node is offline, PoI delegates the block harvesting to another nodePayment across Bitcoin network consumes 100 times the electricity consumed by NEM network [90].

3.12. Proof of Authority (PoAu)

Gavin Wood, the Ethereum co-founder, proposes the concept of PoAu in 2017 [95]. It is suitable for permissioned network where the identities of participants are known and reputed [96]. A set of trusted nodes called authorities, with unique ids and honest reputation, perform transactions of clients by reaching consensus; thus, they act as validators [97]. It is different from PoS where the user stakes certain amount of coins, as in this case the block validators put their reputation at stake [95].

The advantages of PoAu include increased efficiency, transaction, and scalability. Less decentralization and publicly known identity of validator are the known limitations of this consensus [95]. Its applications can be seen in Ethereum Proof-of-Authority Consortium in Microsoft Azure [98], Ethereum's Kovan testnet, VeChain Thor blockchain network [99]. There has been collaborations made with Kovan testnet for applications in the field of Digital Asset Management (MelonPort), Financial Instrument Insurance (Nivaura), Asset Tokenization Platform (Digix), and Decentralized Energy data application platform (GridSingularity) [100,101].

3.13. Proof of Weight (PoWt)

PoWt uses a value called weight that represents the contribution of the user to the network. This value can be compared with the coins owned by user in PoS. The only difference is that in PoWt, any value can be used as "weight" and not only the coins owned by user as in PoS. Hence, the user with maximum weight has a greater chance of discovering the next block [102]. PoWt is like an umbrella definition that holds a few other consensus algorithms under it. Proof of Spacetime, for instance, uses IPFS data to be used as weights. Proof of Reputation is another example that uses this concept of PoWt [103]. The implementation of PoWt can be seen in cryptocurrencies like Algorand and Filecoin [104].

3.14. Proof of Reputation (PoR)

PoR is used in permissioned blockchain as reputation is linked with identity. Since this is a permissioned network, every participant has a copy of public keys of other participants in the network and the private key is stored locally. In this consensus, a distributed ledger of reputation is maintained by each participant. Unlike PoW, where the miners compete for validating next block, the PoR allows the most trusted or reputed node to validate the block and publish it resulting into an increment in the reputation of the validator node [105]. Companies, not individuals as in PoW, are chosen as validators. This puts the reputation of the different companies at stake creating a better decentralization, transparency, and accountability. Reputation at stake also leads to reduction in bribery and vulnerability [106]. As this is a permissioned network and the number of validators is small, the transaction per second can be increased by increasing the block size [106]. It produces reduced energy consumption as there is no need for mining [106,107]. PoR has been implemented in GoChain for improvement of smart contracts and DApps [106,107].

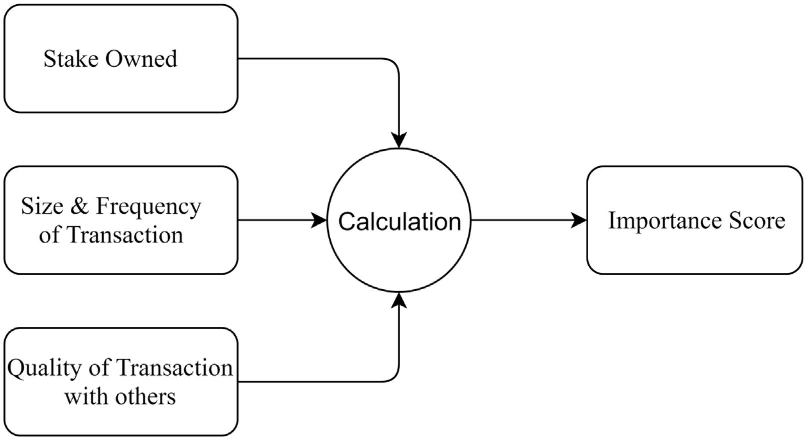

Stake Owned
Size & Frequency of Transaction
Quality of Transaction with others
Calculation
Importance Score

Fig. 9. Calculation of score of the importance in PoI.

3.15. *Proof of Personhood (PoP)*

PoP is a proposed consensus approach in which the accountability of user is maintained by mapping virtual identity of the user node with physical entity and preserving the anonymity of the user [108]. There are three stages in this approach: pseudonym party setup, pseudonym party operation, and pseudonym party end. A group of independent persons called as organizers (also considered co-nodes), in charge of an independent server, organize a pseudonym party. The organizers select a group of Observers, a different group of people, who record the party and produce a video file to maintain transparency. Following this, organizers create a configuration file which contains following information: place of Party, date of Party, start time and end time of party, time at which tokens issuing will be started, video file and organizers Public Key, etc. One assumption made for success of this system is that in which the attendees of the party trust at least one organizer [108]. After a testing procedure, the party is advertised, and configuration file is published from where it can be downloaded by attendee. The attendee must create his own public and private key. The public key is generated using the configuration file and is called as ephemeral public key. This is the setup phase for pseudonym party. The next phase represents operation during the pseudonym party. Attendees enter the party, and no one can enter after a certain period called as barrier point of time but, anyone can leave the party. This is followed by PoP token issuing process where attendee's ephemeral public key is stored to mark his/her attendance with an ink stamp. In the end, the party transcript, containing the hash of the video file recorded by the observer and the list of attendee's public key, is signed by the co-nodes. This step concludes the end of the party. Party transcript is handed over to all attendees and upload at the end of party. It is used for authentication of the.

Two challenges are associated with PoP. One, it must ensure that specific public private key pair are used for only this service and not any other service. Second, it must ensure that the token gets generated at the pseudonym party. PoPCoins, a cryptocurrency, uses PoP. It is based on notion of one PoP token and one vote principle [108]. Possible applications with this mechanism are: secure governance and e-voting, making social media resistant (against trolls, bot propaganda campaign and fake news), and a fair decentralized approach [109].

3.16. *Proof of Learning (PoL)*

PoL is a proposed consensus mechanism created with an aim to validate a transaction and store the experiments and machine learning models on a distributed database. The algorithm is being used for validation of transactions in WekaCoin blockchain [110].

The validation process in this consensus involves three roles: supplier, trainer, and validator. Supplier publishes the task in the form of machine learning competitions and this step is called as task publication transaction which is digitally signed by suppliers' private key. It contains publication timestamp, training dataset, reward, performance metrics, evaluation script, baseline performance score, and cryptographic hash of the test dataset. The supplier must send a reward amount for the winner along with a hosting fee for the task to an empty address. On completion of training period, test data set is uploaded by the supplier. Trainer participates in the competition and tries to form appropriate model for the task. It submits the model by a step called as model transaction in which the model created is signed using trainer's private key and is published over the blockchain. Validator performs the evaluation of the model based on the test data. The model that can be evaluated in terms of performance metrics like F-score, accuracy, and etc., based on which a ranking is made for the models. The model with this value above a certain level receives the reward [110].

3.17. *Proof of Federate Learning (PoFL)*

PoFL uses a reverse game-based data trading mechanism and a privacy-preserving model verification mechanism [111]. Reverse game-based data trading provides security against training data leakage. The verification mechanism verifies the accuracy of a trained model with privacy preservation of the task requester's test data as well as the pool's submitted model. In PoFL, the framework considers the problems such as image recognition and semantic analysis as tasks on a platform by the requester. It also includes the corresponding rewards as incentives for mining. The amount of reward can indirectly reflect the importance and urgency of a task. PoFL chooses the task with the highest reward as the current problem that should be solved by miners as a PoW to reach a consensus. For multiple tasks with the same reward, PoFL selects a task based on first come first execution basis.

3.18. *Proof of Individuality (PoInd)*

Sybil attack is a problem in cryptocurrency [112]. PoS-based networks are more prone to such attacks than PoW-based network. In the proposed concept of PoInd, the users across the network are segregated into sets with each set containing few members (five or more) only who partially communicate with each other within that set to prove their existence [113]. This is followed by signing of each other's identity and its verification to make sure that a user does not become active in more than one group at the same time. Upon verification, this identity represents each unique human being, and this will be known to everyone [112].

3.19. Proof of Luck (PoLu)

PoLu encourages the use of Trusted Execution Environments (TEEs) instead of using ASICs. For success of this algorithm, the participants must maintain the blockchain. Each participant has TEE (example Intel SGX) implemented on its CPU and the program should be able to generate a random number (number between 0 to 1). The miner with the highest value of this random number is considered as the winner and its block is added to the block chain. Hence, it called as Proof of Luck as the number generated is random. Thus, the nodes mine blocks with their lucky number and append it but only the largest lucky number blocks are added to the main chain. The blockchain built upon this TEE-based PoLu is ensured to maintain liveness and persistence [114].

3.20. Proof of eXercise (PoX)

The proposed method makes the miner to solve a real-world based computational problem instead of energy consuming cryptographic puzzle (as in case of Proof of Work). This real-world based problem can be considered as a matrix like problem. While some approaches like PoS and variants introduce a new form of mining process. There are also other approaches which try to solve the issue leaving and creating an infrastructure that is vulnerable to new risks. There are also other approaches (example: as used by Prime coin) which use a better problem replicating the real world issue instead of using the puzzle problem as used in PoW [115].

3.21. Proof of Vote (PoV)

This consensus mechanism is suitable for a consortium network where business or enterprise related companies can come together for transaction purpose. The main idea behind this procedure is that the members of each company vote for a member within the company and one with highest vote gets validated as the validated node. This validated node represents the company and is called as commissioner. Each company chooses its commissioner. The role of this commissioner is to verify and forward the blocks generated by an entity, who has the same role as that of a miner in PoW, called as the Butler. The butler produces a block within a certain time else the permission is transferred to the next Butler. The commissioner gives his assent to the validity of a block by encrypting the block header and returning its signature to the butler. On receiving a certain number of signatures from the commissioners within a definite time, the butler considers the block to be valid. The butler calculates a random value R. This value is written to the block and the butler signs it to prove that it is produced by the given butler. It is done within a certain time. This ends a round of consensus. Each round generates this random value R and any butler with this value is given the right to generate new block. Hence, the production of block is done by the butler and verification of the block is done by the commissioner [42]. A procedure for assigning roles is also proposed. Each commissioner recommends and votes for the butler candidates from the ordinary user in the outer network. A commissioner can also recommend himself to become the butler candidate. To maintain high standard, butler is paid high and its work is under the supervision of the members of alliance. Thus, there are four roles in this consensus mechanism: Commissioner, Butler, Butler Candidate, and Ordinary User. Based on these roles, a commissioner team and a butler team is formed. Thus, there is a separation of voting rights and execution rights in the algorithm. Voting rights of the users helps to gain de-centralization as they choose the leader while the executive rights are held by these voted members who perform the validation task [42]. The advantages linked to this mechanism are security guarantee, transaction finality, low power consumption, and no fork. However, there is one drawback as it gives more priority to security over performance [42].

3.22. Proof of Fair PoW system based on rating of user computing power (PoFPoW)

The goal behind this consensus algorithm is to create an equal opportunity for all users to solve a computation problem for generating a block. This maintains fairness in the network. In this consensus, the difficulty of generating a block depends on three criteria: miner's computing power, number of blocks generated by that miner, and participation time for miner. Miner's computing power can be measured in terms of hash rate. Total Rating Value (TRV) is used to represent the rating given to a user based on which the difficulty is linked for a miner. TRV is the sum of the reference rating value and variable rating value. The reference rating is calculated based on the computing power of that miner while the variable rating value is calculated using the number of blocks generated by that block and the participation time of that miner [116].

3.23. Proof of Majority (PoM)

This a proposed consensus algorithm for a private blockchain environment. In this approach, all the miners participate together to maintain a common blockchain by creating and converging the blocks in the blockchain at the same time. Unlike PoW, there is no need to calculate energy consuming hash. The goal of this algorithm is to increase security, improve performance, and reduce energy wastage [117].

3.24. Proof of Lucky ID (PoLID)

The problem of the vast computation in PoW can be solved using this consensus. It uses two hash calculations per node. This reduces the computations drastically [118]. There are two lotteries involved in this proposal: "Omikuji" and "Draw". To prevent prediction, the $Lucky_{ID}$ of verifier candidates is also added as input along with the $Lucky_{ID}$ of the verifier. An Omikuji is carried out by the nodes (1 to n) and the winner transmits the block along with his $encrypted_{ID}$ and the $Lucky_{ID}$ of all the participating nodes (1 to n). Nobody should own resources enough to obtain wining probability or predict the amount of computation required. The key idea is that using multiple digital signatures input in a lottery, the winning probability cannot be controlled, and the results of the lottery can also not be predicted. This forms the basis of this consensus [118].

3.25. Parallel Proof of Work (PPoW)

In this approach a puzzle is solved by all the miners together instead of a single miner. This is called as parallel mining. The key idea is that only the nonce value differs with the miners while the transaction data remains the same for different miners. This responsibility is taken by the manager. A manager ensures that no two miners use the same nonce value but, all the miners use the same transaction data. For each epoch there is a different manager. Thus, the validity of manager remains only for a certain block for which it is responsible. Miner who solves a block becomes the manager for a block that comes after the next block. If miner solves the block number n, then it becomes the manager for $(n+2)$th block. For the genesis block (block at the start without any transaction), there is no manager while for the first block, manager is chosen randomly among the miners [39]. For each block mined, the reward splits between the miner of that block and the manager of that block in 40:60 ratio respectively. Thus, the miner who acts as a manager receives two rewards: 40% transaction fee for $(n-2)$ block as a miner of that block and 60% transaction fee for block n as a manager of that block. The proposed system is a ring like structure where the nodes are connected to the manager in addition to each other [39].

3.26. Proof of Property (PoPr)

This research work mentions two issues linked with the present blockchain technology. First, any new member requires to download the whole blockchain during initial period. This creates a delay in the participation. Second, those who do not have sufficient space to download, this blockchain fails to participate in the creation of new block due to inadequate link resulting in their inability to validate the incoming transactions. The proposed approach aims to solve these two problems. The term ‘property’ here means the coins owned by a user of certain address and ‘proof of property’ means that a user that performs a certain transaction has ownership of this property [119].

3.27. Proof of Sincerity (PoSin)

The problem of monopolization in some blockchain algorithms gives benefit to the users with high computation power and resources. To avoid such issue, this approach of sincerity is proposed [120]. Such a method also increases security level of a distributed ledger. In the proposal, the definition for a unit of sincerity is given to be the level that requires to compute hash code with 1 bit of leading zero by consumption of their own resources by the node. Mining becomes successful if there is at least one miner with a certain expected sincerity level with a predetermined number of leading zeros. The advantages of using this method are improved chances of participation for miner with small computing power, configuration against 51% attack, and no orphaned block problem. The limitations are that PoSin requires comparatively more energy than PoS and extensive accumulated resource is still required for sincere work to be performed.

3.28. Proof of Relevance (PoRel)

In this approach, the next block that gets added to the blockchain is determined by calculating the relevance of that block. Hence, for a block to get added, it must be most relevant to the previous block. The proposed algorithm is believed to be improvement over PoW as this approach shows more resistance against the 51% attack and localization problem. It is recommended that a separate methodology should be adopted for determining the proprietary information. The pair of public private key for a block, used to create a proprietary information (proprietary key) for that block in addition to signature purpose, is used to generate relevance of that block. There is a use of two chains, key blockchain and authentication blockchain, instead of a single blockchain [121].

3.29. Proof of Witness Presence (PoWP)

PoWP is based on the concept that the presence of witness is essential for maintaining its authenticity. The user needs to prove its presence at a certain location at a certain activity by performing some action; this in turn provides the user with some incentives for taking part in this activity. The idea is initially proposed for better democratic environment for better decision making. It also involves use of artificial technology and self-governance at various steps for improving the decision making ability. GPS (Global Positioning System), LPWAN (Low Power Wide Area Network), mobile cellular network, and P2P ad hoc network are the discussed methods that can be used by citizens to prove their location and over this proof of witness can be built [122].

3.30. Proof of Approval (PoApp)

This is a permission-less and stake-based protocol. It encourages the presence of nodes on cloud rather than their premises. The members or the nodes in the network are called as parties. Each party contains a public–private key pair, a stake in the network, and its ’Coin Roll ID’. The unit for stake is coin and a roll is bigger unit for stake that contains larger number of coins. There are three main steps: block creation, block approval, and epoch approval. Rewards are also provided for these activities. The winning block (only one) gets the block creation award along with the transaction fee. Awards are also given for block approval and epoch approval [123]. Blocks are produced at discrete time intervals called as slots. Some rolls are selected for each slot and their owners are allowed to create blocks. Each slot produces at most one block that gets placed in the blockchain. This also means that it can result into an empty slot as well. Based on this slot, the clocks of the members of the network is synchronized. With a new slot, there may also be a change in the number of nodes in the network. An epoch is bigger interval that contains n number of slots in a continuous manner. It can be deduced that an epoch may have more slots than blocks created due to the presence of empty slots. Nodes, with a minimum number of stakes, approve the block and place it in the blockchain. The block validation predicate is used to validate these blocks. If the approval stake of the candidate block of a node is greater than a certain value, then an approval block is broadcast across the network. Epoch approval involves approving a block from that epoch and broadcasting it across the network [123].

IoT and Proof-of-X. The future of IoT is potentially limitless. Advances to smart vehicles, industrial infrastructures, ubiquitous and pervasive computing, and smart homes-cities and factories get acceleration through network agility, inclusion of Artificial Intelligence (AI), capacity to deploy, configuration to automate and orchestrate, and ensure the security in diverse use cases at a large scale. Along with the potentials of connecting billions of devices simultaneously, IoT also leverages the huge volumes of transactional data for automating diverse business processes. The on going 5G technology expects to gear up the growth of IoTs. IoT connects various low-complexity devices which are delay-tolerant; the devices focus on being low-complexity and power efficient. Therefore, the futuristic developments for decentralization and distributed operations in IoTs need efficient protocols. As a result, the consensus algorithms for blockchain decentralization in IoTs must be energy-efficient and less complex to be configurable in IoT devices. With an anticipation of a 6G standard in 2028 and commercialization of 6G by 2030, consensus algorithms for IoTs must be more sophisticated in terms of complexity and energy consumption. We show advantages, disadvantages, and the IoT applicability for each Proof-of-X consensus in Table 1.

4. Consensus based on PAXOS

Paxos is a family of protocols for solving consensus in a network of unreliable nodes. Leslie Lamport explains a simplified explanation of PAXOS algorithm [128–130]. There are three roles in PAXOS: proposer, acceptor, and learner. Proposers are those who propose value for which consensus is required. Each proposal has a proposal number linked to it. Proposers send two kinds of requests to the acceptors: *Prepare Request* and *Accept Request*. Acceptors are those who contribute to reach consensus and perform the task of accepting or ignoring the proposal (sent by proposers). Acceptors do not accept the message requests from the proposer if the proposal number is less than the current or same proposal number is used in previous round of PAXOS. Learners are those who learn about the consensus value and accept it. A single consensus value reaches in each PAXOS round. Thus, it requires another round to get another consensus value [128]. Fig. 10 represents the orientation of different roles in this algorithm. It shows the connections among the roles. In practice, a single node may run proposer, acceptor, and learner roles. It is common for Paxos to coexist with the service that requires consensus (e.g., distributed storage) on a set of replicated servers. Each server takes on all three roles rather than using separate servers dedicated to Paxos.

Table 1
Advantages and disadvantages of the consensus protocols based on Proof-of-X.

Consensus algorithm	Ref.	Advantage	Disadvantage	IoT applicability
Proof of Work	[34,36,37,40]	Prevents the DoS attack or spammers, Provides security to entire network, Prevents double spending attack possible in distributed systems	Energy and resource consumption, 51% attack, wait for finality, mining dependency vs computing power	Mining is not suitable for IoTs, lightweight mining or intelligent mining should be explored
Proof of Stake	[43–45,49, 50]	Consumes less energy than PoW, no need to use special hardware	Monopolization of control, coin hoarding	Usable for IoTs, dynamic stakes create problem as IoT devices use on-the-fly
Proof of Burn	[45,53,54, 124]	Avoidance of computational resources, long term commitment of the users, decentralization, less energy consumption than PoW	Burning to coin causes resource wastage, coin hoarding , , and intentional control	Not suitable for IoTs due to the burn process, alternative resource disposal should be explored
Proof of Space	[58,60]	Energy efficient as it uses low power hard drives instead of ASICs, CPUs and GPUs, Increased scope for decentralization	scaling, probabilistic monopoly with large spaces	Applicable for IoTs, the large space problem can create monopoly in heterogeneous devices
Proof of Stake Velocity	[64]	Used an alternative to PoW and PoS, introduces new form, reduces wastage of mining, reduces the mining race	Social interaction as a measure of currency, effects of inflation	Applicable for IoT, needs proper authentication and access controls for social interaction
Proof of Activity	[66,67]	Secure than PoS and PoW against 51% attack, DoS-resistant, decentralization	Resource requirement, chances of monopoly with higher stakes for trying to double sign the transactions, trade-off between the energy and monopoly	Not suitable for IoTs due to energy consumption
Delayed Proof of Work	[125–127]	Reduces the amount of computation required as in case of PoW, secure against 51% attack	Not considered as a consensus algorithm but acts as an additional layer of security	Applicable for IoTs due to less computation, additional consensus required to ensure decentralized decision
Delegated PoS	[76–78]	Faster processing than PoW and PoS, energy efficient than PoW, detection of malicious action possible	More susceptible to attack because few people are in charge, less decentralized	Applicable to IoTs theoretically, needs proper validation
Proof of Elapsed Time	[82]	Collision resolving, consumes less energy, increased efficiency than PoW	Security of PoET depends on underlying computing hardware that may lack reliability and can be prone to attacks	Physical Unclonable Function (PUF)-based integration required for IoT applicability due to security issues
Proof of Space and Time	[86–88]	Eco-friendly, more decentralized	The prover has to read entire table for generating a valid response to a challenge	Applicable for IoT with optimization of read-cycle of table
Proof of Importance	[89–92]	Consumes 100 times less energy than that in Bitcoin network	A user may send transactions back and forth continuously to gain reward and increase its chances	Energy efficient for IoT but not secure due to transaction dangling
Proof of Authority	[95–97]	Higher transaction rate, more efficiency, higher scalability, more secure	Tendency for centralization, validators are publicly known so there are chances of compromise	Applicability in IoT requires strong authentication and access control, privacy with pseudo-identity must be explored
Proof of Weight	[102–104]	Energy Efficient	An umbrella definition that holds other types of consensus within it	Application of this consensus in IoT needs validation
Proof of Reputation	[105–107]	Increased decentralization, increased transaction speed, hard to bribe, reduced energy consumption	Implemented as a prototype model only	IoT application needs validity with an IoT framework
Proof of Personhood	[108]	Increased accountability of members along with maintenance of anonymity	How to ensure that the public private key will be used for this specific purpose only, how to make sure that token gets generated at pseudonym party only	Needs proper privacy measures and secure key management for heterogeneous IoTs
Proof of Learning	[110]	Alternative for PoW , storing the learning models and experiments in a distributed and verifiable database	Inability to provide continuous supply of tasks for validation purpose, privacy associated with the datasets that are supplied for validation	Applicable for smart IoTs, collaborative learning on distributed parameters and dataset privacy must be explored further

(*continued on next page*)

Table 1 (*continued*).

Consensus algorithm	Ref.	Advantage	Disadvantage	IoT applicability
Proof of Federated Learning	[111]	Homomorphic encryption based label prediction, low energy consumption	System-level heterogeneous characteristics make issues such as stragglers and fault tolerance	Applicable for smart IoTs, label prediction accuracy can be increased and must be explored with iteration convex for the optimal decision
Proof of Individuality	[108,112,113]	Members of the network are verified	The virtual live stream used for verification can be compromised	Applicable for IoT with lightweight verification process at the initial stage
Proof of Luck	[114]	Liveness and persistence, energy efficient, low transaction latency, Deterministic confirmation time, deterministic mining power, permission less and scalable	An assumption is made that the adversary is not able to break any cryptographic primitives with non-negligible probability	Applicable for IoTs, complexity and energy consumption analysis required
Proof of Exercise	[115]	An alternative for PoW algorithm	The complexity of Proof of Exercise is much higher than PoW	Not applicable for IoTs due to high complexity
Proof of Vote	[42]	Controlled security than PoW, low power consumption, one block confirmation to achieve transaction finality, no fork, low delay transaction and verification time, low transaction latency	Sacrifices performance to maintain security	Applicable for IoT with optimized voting scheme
Proof of fair PoW based on Computer Power	[116]	Tends to solve the problem of decentralization of PoW	Some may consider it as a biased method	Validation for practicality required
Proof of Majority	[117]	Reduces energy wastage, improves performance, provides high security in a private blockchain, does not require hash calculations	Requires controlled and private blockchain environment only	Not suitable for heterogeneous public networks
Proof of Lucky ID	[118]	Drastic reduction in computation in comparison to PoW, consensus achievable in less time compared to PoW and PoS	Random number need to be true random, else monopoly control may exist	In an environment of billions of devices, random number generator must be secure enough and configurable for low complexity IoT devices
Parallel Proof of Work	[39]	Increased transaction speed, fairness to miners, more decentralization	Not evaluated in real time network	Not significant for IoTs due to mining
Proof of Property	[119]	Reduced need for hard drive space for participation, decreased data overhead	Considers only one state (like that in bitcoin) of balance of the addresses, not resistant against forks	Space limitation may exist in IoT framework as devices are resource constrained
Proof of Sincerity	[120]	Improved chances of participation for miner with small computing power, resistant against 51% attack, No orphaned block problem	Requires comparatively more energy than PoS	Not suitable for IoTs due to high energy consumption
Proof of Relevance	[121]	More resistant to 51% attack than PoW and localization problems	Not practically implemented yet	Validation required for practical implementation
Proof of Witness Presence	[122]	Proposal for better governance, democracy, and smart city implementation	Witness calculation and validation	Applicable for IoT, witness verification required for heterogeneous IoTs
Proof of Approval	[123]	No need for consumption of energy, instant finality	Need for additional storage in comparison to other consensus protocols	Applicable for IoT if storage limitations can be eliminated

PAXOS uses two phases. Phase 1 is called as the promise phase and phase 2 is called as the commit phase. In phase 1, the proposer sends a request with a proposal number n to each acceptor node. Proposer receives a response from these acceptors; the response contains information that includes confirmation and a proposal number (i.e., the biggest number seen by acceptor). If majority of the acceptors reject the proposal of the proposer, then it gets updated with the recent highest valued proposal number. In case of acceptance by the majority of acceptors, a confirmation message (accept message or promise) is sent to the proposer. This confirmation acts as a promise that the nodes do not accept any proposal number less than the one sent by the proposer. In addition to this, the acceptor nodes send back the proposal with the highest number less than n that has been accepted. When a certain node within the set of acceptor nodes fails or becomes malfunctioned for a certain round, then after its recovery this step helps the acceptor node cope up with the current state of the network [128,131]. When a majority of responses from the acceptors is accepted, the proposer node issues the proposal associated with number n or value v, where, n is the proposal number accepted by the majority and v is the proposal number that has the highest response from the acceptor nodes. It can also include a value, in case, no proposal has been reported by the responders. In this step, the proposer issues a proposal by sending a request the proposal be accepted [128]. Phase 2 of this algorithm can be considered as the commit phase. The proposer receives a response to its already made prepare requests from the majority of acceptors. The proposer sends an accept request to each of these acceptor nodes for the

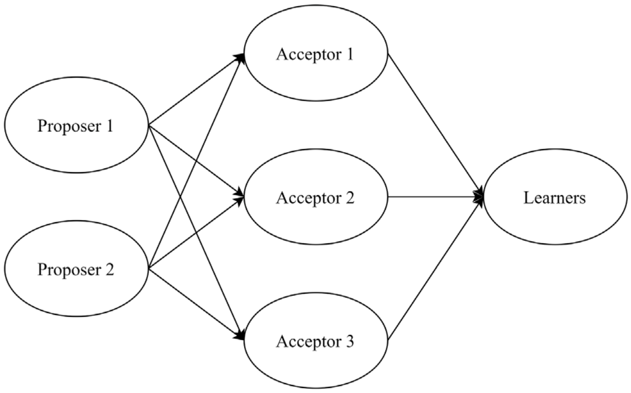


Fig. 10. Representation of different roles in the PAXOS algorithm.

already proposed number n that is assigned with value v; it contains the highest numbered proposal among the responses or any other value in case of any non-reported value from proposal. On receiving the accept request, the acceptor accepts the proposal [128]. This information of acceptance of proposal is passed on to the learner by the acceptors.

Fig. 11 represents the two phases of Paxos algorithm. Proposer sends the 'Prepare 5' message to acceptors. If the proposal is accepted by the majority, a confirmation or promise is sent to the proposer by the acceptors. Here we show one acceptor to represent the group of acceptors. Another 'Prepare 3' is sent by the proposer that has a proposal number less than 5. Hence, it is ignored. In the second phase, the proposer sends request 'Accept Request 5' to the acceptor. The acceptor passes the value associated with the accepted proposed number 'Accepted Value' to the proposer and the learner [1].

The drawbacks linked with PAXOS algorithm include the understandability of its working and the purpose of each phase [13], inefficiency of two rounds for a value to be chosen, and incompleteness [13]. Lamport shows a complicated proof which makes it difficult to understand or implement [132]. An application of PAXOS can be seen in Apache Cassandra, Reddit storage, Netflix backend database, Google's megastore, and Chubby lock services [133–136].

4.1. Multi-Paxos

Multi-Paxos can be built from basic Paxos by carefully adding slots [137]. The algorithm of multi-Paxos consists of repeatedly executing the two phases. Therefore, it has four phases: 1a, 1b, 2a, and 2b. The 1a phase follows the basic Paxos with a small change i.e., it replaces the ballot number b in basic Paxos by the proposer p executing this phase. The messages of 1a is not having any specific receiver; thus, messages can reach to all the processes. However, it maintains the condition of at least one quorum in the receiving set. In 1b phase, acceptor considers triples of ballots, set of values to be proposed, and slot values. In 2a phase, the bloating of a single value in basic Paxos extends to use a set of pairs of slots and the proposal values. The operation of finding the value with the highest ballot in the basic Paxos is performed for each slot. The ballots are shared and changed for all slots; slots are paired with values dynamically, where slots that fails to reach consensus values earlier are also detected and reused. This makes the multi-Paxos significantly better than basic Paxos. In 2b, the acceptor replies with a set of slot and values for each slot. The main objective of multi-Paxos implies the same leader is handling multiple client requests without running the leader election repeatedly.

4.2. Cheap-Paxos

Lamport and Mike have shown a variant of Paxos, called as Cheap-Paxos [138]. The authors use a system of $F + 1$ main processors and F auxiliary processors. The main processors act as the servers in a distributed state machine implementation. The auxiliary processors perform actions only in the event of the failure of the main processor, after which the main processors continue to operate the system by themselves. Cheap Paxos uses dynamic Paxos where the set of acceptors and the quorums are determined by the state machine itself and state machine performs reconfiguration. The key contribution in cheap Paxos is that a leader can send its "1a" and "2a" messages to a quorum of acceptors. As long as all acceptors in that quorum are working and can communicate with the leader and the learners, there is no need for acceptors not in the quorum to do anything. Cheap Paxos uses the Paxos consensus algorithm to choose commands and follow the same safety properties directly from the basic Paxos consensus.

4.3. Fast-Paxos

In general, communication pattern in Paxos consensus follows the flow of messages as: proposer to leader, leader to acceptors, and acceptors to learners. In fast Paxos, the proposer sends its proposal directly to the acceptors bypassing the leader [139]. This saves one message delay and one message. Fast Paxos uses round numbers that is partitioned into two types: fast and classic based on round number. The rounds follow basic Paxos with some changes. In a fast round i, if the coordinator picks any proposed value in phase 2a, it sends a special phase 2a message to the acceptors. When an acceptor receives this message, it may treat the value as of a normal round and then follows the basic Paxos.

4.4. Byzantine-Paxos

In this consensus, the author considers f fake acceptors to byzantinize the basic Paxos [140]. It uses a variant of basic Paxos by introducing 1c phase and changing the 2a phase. Phase 1c uses the 1b messages from a quorum of acceptors, the leader chooses a set of values which are safe at b and sends a 1c message for each of those values. In the modified phase 2a, the leader sends a 2a message for some value for which it has sent a 1c message. The byzantinizing of this variant is called as Byzantine-Paxos (BPCon). There is no explicit 2a message or Phase 2a action in BPCon. Instead, the acceptors cooperate to emulate the sending of a 2a message. The ballot-b leader requests that a Phase 2a action be performed for a value v for which it has already sent a 1c message. On receiving the first such request, an acceptor executes a Phase $2av$ action, sending a ballot-b $2av$ message for value v, if it has already received a legal ballot-b 1c message with that value. An acceptor executes a ballot-b phase $2av$ action iff it has received the necessary 1c action and has not already sent a ballot-b $2av$ message. Acceptor accepts only legal 1c messages. Byzantine-Paxos also have another variant as Fast Byzantine-Paxos [141].

4.5. Disk-Paxos

Disk-Paxos is an extension of Paxos synod algorithm. Each processor starts with an input value [142]. An extra phase includes the recovery of a processor with a new input value. Different processors use different ballot numbers. A processor uses a ballot in two phases: first, trying to choose a value and second, trying to commit that value. The choice of the value occurs in the transition from phase 1 to phase 2. The value is committed and can be output when the processor finishes phase 2. To execute the algorithm, a processor maintains a record consists of three components: (i) the current ballot number, (ii) the largest ballot number for which the processor enters phase 2, and (iii) the value processor tries to commit in the largest ballot number.

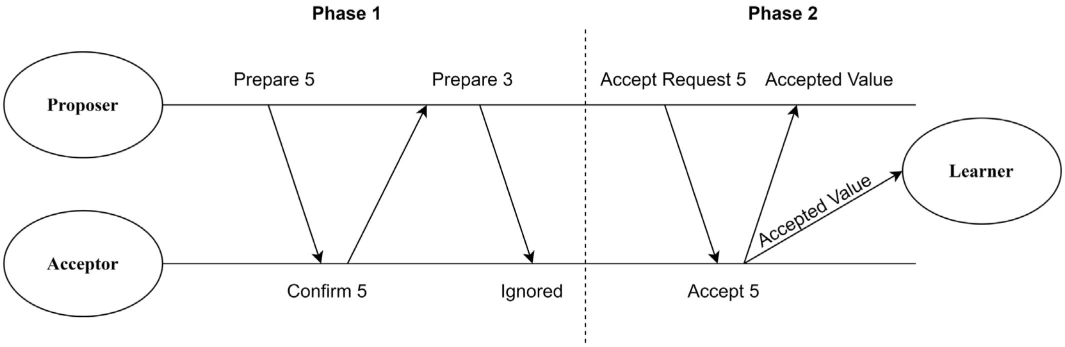


Fig. 11. Diagrammatic representation for Paxos algorithm.

4.6. Vertical-Paxos

In this variant of Paxos, reconfiguration can occur in the middle of reaching agreement on an individual state-machine command. Vertical-Paxos uses an auxiliary configuration master that facilitates agreement on reconfiguration [143]. Vertical-Paxos follows the traditional Paxos; however, two major changes make them separate.

- *Read/Write quorums*: Practical protocols use primary-backup structure for better system-wide resilience. Vertical Paxos achieves primacy-backup structure by distinguishing between read and write quorums.
- *Auxiliary configuration master*: It considers a separate configuration master called as auxiliary master. It allows the set of acceptors to change within each individual consensus instance. The master determines the set of acceptors and the leader for every configuration.

4.7. Egalitarian-Paxos

Egalitarian-Paxos or EPaxos is based on two prime ideas [144]. First, most of the commands do not interfere with each other. Therefore, it is not necessary to use a consistent ordering for command execution. Second, EPaxos does not check instances for the commands; it determines the ordering of the instances in the process of choosing commands by using attributes to each command. Once a command is committed, all the non-faulty replicas have a consistent view of the attributes for that command. It then follows the execution of the command in the same order relative to other interfering commands.

4.8. Mencius

Mencius is a consensus protocol for general state machine replication that tolerates crash failures [145]. Each instance of consensus is assigned to a coordinator server. The coordinator is the default leader of that instance. Mencius follows round-robin for coordinator assignment. A server proposes client requests immediately to the next available instance it coordinates without waiting for other servers. Thus, it reduces latency. Mencius rotates the coordinator assignment. This helps for the servers to propose directly.

4.9. Stoppable-Paxos

Stoppable Paxos uses the same variables and sends the same messages as Paxos [146]. An active leader performs a Phase1a(b) action for a suitable ballot number b. If the leader finds that a stopping command stp is chosen in some instance i, the leader performs Phase2a actions for lower-numbered instances as before. However, to ensure that the state machine stops when it should, the consensus must ensure that the leader does not perform a Phase2a action for any instance greater than i if the stopping command actually was chosen in instance i.

4.10. Paxos for RDMA networks

APUS's consensus protocol has three main elements [147]. First, it uses a Paxos consensus log. Second, threads of a server program run on the leader host (or leader threads). APUS uses the inbound socket calls (e.g., recv()) of these leader threads and invoke consensus requests on these calls. We denote the data received from each of these calls as a consensus request (i.e., an entry in the consensus log). Third, an APUS internal thread runs on every backup (or backup threads) that agrees on consensus requests. The leader enables the first and second elements, and backups enable the first and third elements. This thread's consensus request has four steps. The first step executes the actual socket call, the second step is a local preparation, including assigning a view stamp for this entry in the consensus log, allocating a distinct entry in the log, and storing the entry to a local storage. In the third step, each leader thread concurrently invokes a consensus and writes the log entry to remote backups. This step is thread-safe because each leader thread works on its own distinct entry and remote backups' corresponding entries. In the fourth step, the leader thread polls on its reply field in its local log entry to wait for backups' consensus replies. It breaks the poll if a number of heartbeats fail. If a majority of replicas agrees on the entry, an input consensus is reached, the leader thread leaves this recv() call and proceeds with its program logic.

4.11. Stellar Consensus Protocol (SCP)

SCP is proposed in year 2015 by David Mazieres with an aim to remove the centralization probability, middlemen mismatch in global transactions. Decentralized control, low latency, flexible trust, and asymptotic security are ensured in this consensus [125]. Stellar consensus allows each validator to select its own list of validators and this list is called as Quorum slice. These quorum slice of various members overlap to form a quorum for a network. Quorum is the minimum number of nodes required to reach consensus.

There are two steps for implementation of SCP: Nomination Phase (phase 1) and the Balloting Phase (phase 2). In the first phase, each node nominates a value or does not give a value at all; it cannot vote against any nominated value. This continues till at least one value confirmed is nominated for each slot and propagated across the other nodes so as to converge to a same set of nominated values. Later these values are combined into some composite value (using union, etc.). In the second phase, after the nodes have converged to a value in the previous phase, a ballot is attached to the candidate. Federated voting is used to commit and abort the sets of ballots [127]. Stellar is a payment network that allows financial transactions to take place. Lumen (XLM) is its digital currency that has the minimal role [148]. Mobilecoin is a cryptocurrency that uses SCP as underlying consensus mechanism for synchronizing the transactions. It is designed for faster transactions, privacy, and user-friendly features [149].

Table 2
Advantages and disadvantages of PAXOS and its variants.

Name of Algorithm	Ref.	Advantage	Disadvantage	IoT applicability
PAXOS Algorithm	[1,13,128, 131]	Oldest algorithm for distributed consensus, its incompleteness gives opportunity for potential and improvements	Hard to understand, complex algorithm lacking understandability, incomplete	Extension of Paxos can be applicable with lightweight processes for the proposer, acceptor, and learner
Multi-Paxos	[137]	Using slots, triples for each slots for both proposing and acceptance, no need to run leader election multiple times	In master–slave architecture, gap of commands arises and needs a proper recovery process after detection of master failure	In client–server based IoT architecture, Multi-Paxos can work, leader election can be critical and needs further research for an appropriate process for the election
Cheap-Paxos	[138]	Fault-tolerant, resource utilization for auxiliary processors	No liveliness	The freshness of the key management can be an issue for its applicability in IoT, security can be at a stake
Fast-Paxos	[139]	In a collision, uncoordinated recovery does not create delay, more effective with large number of replicas	Requirement of large quorums	The integrity of replicas in dynamic IoT must be maintained and heterogeneity must be addressed
Byzantine-Paxos	[140]	It supports state-machine replication with tolerate for byzantine faults	Resource consumption, optimization of the number of required replicas and copies needs to be explored	Not suitable for IoTs as resource consumption is high
Disk-Paxos	[142]	Minimal number of disk accesses for a consensus algorithm in presence of failure of any minority of the disks	If a processor fails while in writing, its disk blocks may go in a state in which no value has been written to a majority of the disks leading to shared-memory problem	Shared memory problem raises unwanted resource consumption in IoT devices
Vertical-Paxos	[143]	Works as backup structure, less power requirement for auxiliary master	Synchrony must be ensured for working of configuration manager	For master–slave IoT configuration, load balancing on masters must be addressed
Egalitarian-Paxos	[144]	Decentralized and uncoordinated design, availability, performance stability	Additional round trip tip time with multiple commands' interference. It also increases the total number of messages processed per command, reducing throughput, commit latency may increase in wide-area deployments	Not suitable for IoTs due to increased latency and message overhead
Mencius	[145]	It handles server crash, rotation of coordinator provides load balance	No byzantine support, commit latency is limited by the slowest server	Needs to improve throughput and latency for IoT applications
Stoppable-Paxos	[146]	Consistent, correct, and distributed	No byzantine support, no detection of fake stop command	Not suitable for IoT as intrusion and security breach is easy to occur
Paxos for RDMA networks	[147]	Handling concurrent correctness, fast, and scalable	Leader election on RDMA raises a main challenge because backups do not communicate with each other	Needs optimized and less complex election process for IoTs and requires distributed backups
Stellar Consensus Protocol	[41,72,73,75]	Optimal safety, decentralized control, asymptotic security	reduced performance and communication latency	Not suitable for IoTs due to degraded performance

IoT applicability of Paxos. IoTs are mostly comprised of low-power wireless networks with sensor devices or resource limited embedded systems; they require coordination among the nodes in the network of operations to execute tasks. Failing to agree on a common decision within a time limit leads to system failures. Although Paxos is being used by Amazon S3, Amazon DynamoDB, and Apache ZooKeeper, it is having complications of module integration. Paxos solutions lack behind for the applicability in Iot ubiquitous due to the limited resources of the devices and the high cost (complexity) of the established solutions. Wireless Paxos solves these problems; however, it still requires further exploration to validate the feasibility in IoT domains [150]. We list the various advantages, disadvantages, and IoT applicability of Paxos and its variations in Table 2.

5. RAFT algorithm

RAFT is a consensus algorithm that is an equivalent to PAXOS. In this algorithm, the problem is decomposed into relatively independent sub-problems; RAFT addresses all the major needs of a practical consensus system [59]. A visual representation of RAFT can be seen in [151]. Each server has a state machine and a log. The consensus process in RAFT is broken into three parts: leader election, log replication, and safety. The basic idea is that a leader is elected in the cluster who accepts client request and replicates the log to other servers. The server, at any time, is present in either of the three states: leader, follower, or candidate [152,153]. We show the orientation of these states in Fig. 12.

Phase I of the process is leader selection. At the initialization, there is only one leader and others are followers. Leader is elected by the

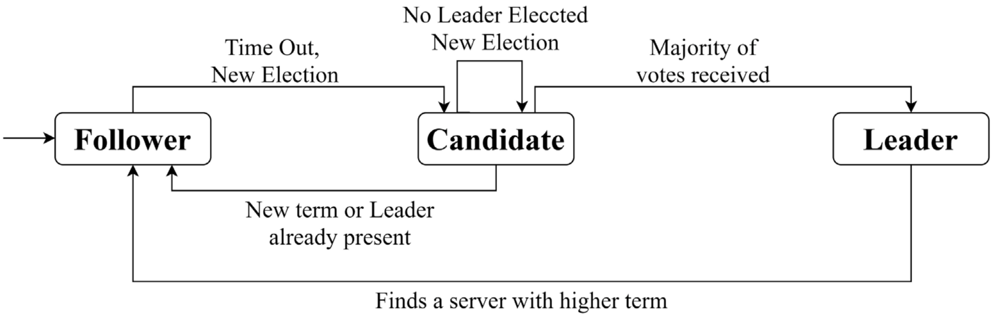


Fig. 12. Representation of state transitions in RAFT consensus.

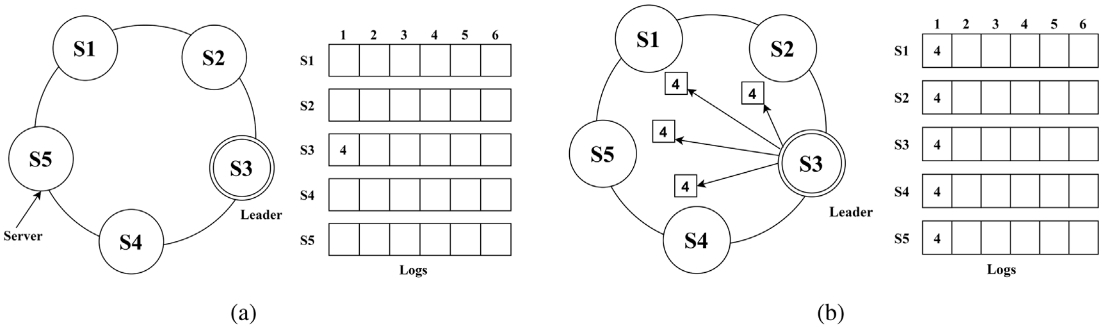


Fig. 13. Representation of committing entries in logs of the servers in a RAFT consensus.

followers during an event of election. The leader issues *AppendEntries* Remote Procedure Calls (RPCs), which act as heartbeats to maintain leadership among the followers and are used to replicate its log [154]. The follower receives regular heartbeats from the leader to realize the presence of the leader. On start-up, each server begins as a follower and continues to remain so as long as it receives RPCs from a leader or candidate. When the follower does not receive the heartbeats from the leader over a certain time (election timeout), then the follower becomes a candidate and issues *RequestVote* RPCs to get elected as a leader. Thus, an election starts for leadership role. Follower increments its current state to become a candidate state before the start of an election. This increment executes when any of the following happens: (i) some other server wins the election and becomes a leader and (ii) the current candidate wins the election and a certain period of time passes without any declaration of a winner for the leader role [154]. Two things happen when the candidate sends *RequestVote* RPCs to other servers. First, it receives majority votes and becomes the leader. On becoming a leader, it sends heartbeats to other candidates and they become followers. Second, if it receives RPC from some other newly elected leader, then it becomes the follower. If a situation arises where none of the above case occurs, then timeout occurs and the election process re-starts. Phase II begins when the Phase I of leader selection is complete. Client sends command to the leader who appends this command to its log. An *AppendEntries* RPC is sent to all the followers by the leader letting them update their logs. Once the leader gets minimum feedback from the followers, the entry is be considered as being committed. The command can be executed in its state machine and the result can be sent to the client. The committed entries are notified to the followers by *AppendEntries* RPCs and the same commands are executed by the followers in their state machines. Fig. 13(a) shows the initialization of committing of entries in the logs of the followers after they receive signal from the leader [13]. In case of failure or crash of leader, the leader election process starts again while in the case of a follower crash/slow, the leader re-tries *AppendEntries* RPCs till it gets successful response. The durability of command is maintained by it being stored in the logs of maximum followers as shown in Fig. 13(b).

The advantages of RAFT protocol are: (i) easier to understand than PAXOS, (ii) easy implementation than PAXOS, (iii) decomposition into smaller problems, and (iv) operational even after failure of minority servers. However, the limitations involve the non-real world assumptions and too much traffic can result in choking of the system [153]. CockroachDB and Apache Kudu use this consensus [155,156].

A variant of RAFT consensus called as AdRaft implements equal sharing for a single leader node to enhance the performance [157]. It also uses an improved vote change mechanism to solve the problem of multiple rounds in the leader election phase. The vote change mechanism uses the vote count state to candidate node. In a particular round of election, candidate node with more votes can capture the votes of candidate node with fewer votes. This avoids competition of generic RAFT among multiple candidate and thus, AdRaft prevents deadlock. It also uses the communication mechanism of apportionment idea in the log replication phase. It utilizes the idle channel resources of follower nodes to synchronize multiple log messages in a round. It helps in improving the throughput of the leader nodes.

***RAFT and IoT applicability*.** Generally, RAFT consensus is suitable for IoT applied for in a private blockchain. However, RAFT faces problems in public blockchain. If an attacker becomes successful in compromising one node in the network, the attacker is able to change its election time out. This situation increases the possibility for a malicious node to become the leader and manipulate the transactions. Besides, IoTs deal with edge data services in various applications, specifically for vehicular networks, multimedia networks, and smart homes. The required transparency in such services is achievable by deploying the integration of Multi-access Edge Computing (MEC) and RAFT consensus. Base stations and gateways can apply this integration to provide efficient and tamper-proof edge data services. However, latency creates a bottleneck due to the excessive load of data services in one area and less load on other areas. In such a case, migration of transactions becomes useful as shown in [158].

6. Consensus based on byzantine faults

There are two types of failures associate with a distributed network: fail stop failure and not fail stop failure [169]. In both the cases, the

Table 3
Advantages and disadvantages of the consensus based of byzantine fault tolerance.

Name of consensus	Ref.	Advantage	Disadvantage	IoT applicability
PBFT	[159]	Transaction finality without the need for confirmations	Detection and removal of faulty nodes, rights of consortium	Unsuitable for IoT applications due to the need for a single view leader, adaptation to the multiple entry points or ordering node is required for IoTs; however, this is the most used consensus for IoTs for security provisions
Byzantine agreement-based consensus	[127]	Fault-tolerant	Prone to sybil attack	Applicable for IoT if sybil can be prevented well in heterogeneous IoTs
XRP Ledger Consensus Protocol	[160]	High scalability, faster, consumes less energy	Need for maintenance of UNLs	Suitable for IoTs
Delegated Byzantine Fault Tolerant Algorithm (dBFT)	[161,162]	High stability, height TPS, single block finality	Less decentralized than PoW	Suitable for controlled and coordinated IoT applications
Reputation Based BFT	[163]	Secure and reliable due to easy detection of faulty nodes, Higher TPS than PBFT, Lower delay than PBFT	Not implemented in real consortium blockchain system	A suitable option as compared to PBFT, needs further exploration and validation for IoT applications
Multi-Signature BFT	[164]	Guarantees safety and liveness in partial synchronous model, high throughput	Need to remove the complexity of communication with cosigning, witness selection method	Not suitable for IoTs due to communication complexity
FPC	[165]	Leaderless, less complex, moving decision threshold	Anti-Sybil protection required	IOTA Foundation is using this consensus for Coordicide version of IOTA, complete validation is pending for IoT applicability
Istanbul Byzantine Fault Tolerance	[166,167]	Message timeout between phases is reduced, scalable	Justification mechanism of proposed values for round changes must be optimized as it has quadratic complexity	Not significant for IoT applications
MBFT	[168]	Node control in consortium blockchain, reduced communication complexity, low latency, and high TPS	Dependency between LCG and HCG can be eclipsed	Applicable in IoT, reduced dependency required from two different nodes

nodes behave flaky. The general-traitor example is very popular to define the byzantine problem. Different generals in different locations with their troop need to reach a consensus on decision to attack an enemy [170]. This is called as Byzantine General Problem. It is previously proved that there is no solution for $3m + 1$ generals problem if there are more than m traitors. This also indicates that byzantine problem is unsolvable if $1/3rd$ or more generals are traitors. We show the advantages and disadvantages of all the available variants of byzantine faults in Table 3 at the end of this section.

6.1. Practical Byzantine Fault Tolerance (PBFT)

The PBFT algorithm prepares replication for Byzantine faults. It offers liveliness and safety [159]. In the above discussed problem of byzantine, the system is assumed to be synchronous. However, in real-life scenario, the systems are not always synchronous. As stated by *Impossibility Theorem*, it is impossible to reach consensus in a system that is pure asynchronous and has at least one faulty node [132]. Fig. 14 represents different phases of this protocol. There are three phases in this protocol: pre-prepare phase, prepare phase, and commit phase.

When a request is sent by a client to the primary the pre-prepare phase starts; the primary sends this message to the secondary after assigning a sequence number to the given request in encrypted mechanism. This message includes ID of the block published by primary, block number, view number, and ID of primary [171]. In prepare phase, the backups (secondary) send the accepted pre-prepare message to all other replicas in the network which accepts the message if certain conditions are met (correct signature, sequence number within threshold and same view number). In commit phase, each node multicasts the commit message to all the other nodes including the primary node. This creates consensus among the system when the primary receives a minimum number of commits $(2f + 1)$ from the secondary ones.

PBFT has certain limitations. First, a BFT requires $3f + 1$ nodes to tolerate only f byzantine faults. PBFT fails to detect and remove these faulty nodes which break the system security [163]. Second, rights of consortium members may be different. Applications of this algorithm include Tendermint [172,173] and Hyperledger fabric [174].

6.2. Byzantine agreement-based consensus

It is prone to sybil attack. In Federated Byzantine Agreement (FBA), this sybil attack is prevented by selecting quorums from the network. Each node is given a choice to pick a set containing other nodes which are considered by that node and sufficient to gossip for entire network [127]. In FBA, the safety and fault tolerance are chosen to be emphasized. If a network experiences any fork, then the network halts and does not continue with the wrong value [126].

6.3. XRP Ledger consensus protocol

This consensus protocol consists of three steps: deliberation, preferred branch, and validation. In deliberation, there is proposal from

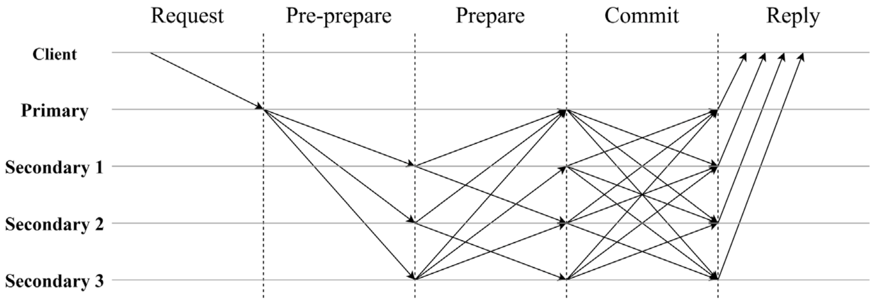


Fig. 14. Representation of phases in PBFT algorithm.

the nodes to add a pending transaction set to the present ledger after it receives support from certain number of trusted nodes. In validation, the ledger that receives more than a certain number of validations from the trusted nodes will considered valid. If there is a case where two or more ledgers qualify for being validated, then the nodes' parent chain determines which ledger to validate [160]. XRP Ledger is used by Ripple in its product called xRapid; it is used by financial institutions. XRP Ledger is open source and no one owns it [175]. XRP shows faster payment and higher scalability than Ethereum and bitcoin. For a payment that takes more than two minutes in Ethereum and more than one hour using bitcoin, XRP takes only three seconds. It is also stable and consumes negligible amount of energy [176].

6.4. Delegated Byzantine Fault Tolerant Algorithm (dBFT)

This consensus is proposed to solve the scalability issue of the PBFT algorithm by adding the feature of dPoS. The algorithm consists of three phases: pre-prepare phase, prepare phase, and persist phase. There are following roles involved in this algorithm: consensus node, ordinary node, speaker, and delegate. The ordinary node is that user who makes a transaction using his wallet. The consensus nodes are the nodes who take part in the consensus activity by proposing a block and voting. These nodes receive transaction made by the ordinary nodes and put it into a memory pool. There are multiple delegates chosen and a speaker is chosen from these delegates (for each round). The speaker broadcasts prepare-request message (containing the transactions from the memory pool) to the delegates who upon verification broadcast the prepare-response message. When the consensus node receives a minimum number (around 66%) response, then a consensus is reached, and the block is published. Once received by a new block, all transactions in that block are deleted from the memory pool. This ends a round of the consensus [161]. In March 2019, an upgraded version called dBFT 2.0 is released [162]. It is used by a cryptocurrency called as NEO [177].

6.5. Reputation based byzantine fault tolerance for consortium blockchain

This algorithm is proposed to remove the drawbacks of PBFT network by introducing a reputation model. It evaluates and score the behavior of all the consensus nodes within the network. If a faulty node is detected to be involved in any malicious behavior, then its reputation is decreased and rights in voting process are also reduced. On the other hand, a node with higher reputation gets greater rights in voting process and increases its chances to generate new valid blocks. Unlike PBFT, a reputation based BFT effectively identifies the faulty nodes [163].

6.6. Multi-signature BFT

This is a consensus algorithm proposed for private blockchain that tends to maintain liveness and safety in the network. The main idea is that apart from selecting a leader for validation of node, a group of participants called as witnesses are chosen. These witnesses sign only one block proposed by the leader. This results in validation of at most one candidate block proposed by the leader [164]. There are three stages in this protocol: proposal, MSigProposal, and vote. In proposal phase, a message called as proposal message (that contains a candidate block, current height, and current round) is proposed to the witnesses to sign. This proposal message is passed on to other witnesses for them to sign. It becomes a MSigProposal message once it contains the signature of all the witnesses. Thus, at a time only one MSigProposal message can be formed. In the vote phase, a node on receiving the valid MSigProposal message broadcasts a vote message. If this node receives $2f + 1$ vote messages before expiration of predetermined timeout $TOcommit$ then the node commits the candidate block and appends this candidate block to the blockchain.

6.7. Fast Probabilistic Consensus (FPC)

FPC within Byzantine Infrastructure (FPC-BI) is a leaderless protocol [165]. It provides low communication complexity and allows a set of nodes to come to a consensus on a value of a single bit. Due to this, IOTA uses this consensus [178]. FPC-BI considers that the nodes are byzantine partially and the attacker may delay the consensus results or stop the consensus finality. FPC-BI uses high probability when its parameters are suitably chosen. A special feature of FPC-BI is that it makes use of a sequence of random numbers. Either a trusted source or some decentralized random number generating protocol generates these random numbers. These random numbers work as "moving decision threshold"; thus, it cannot be predetermined by the attacker and eclipsing is avoided. Therefore, it maintains overall trustworthiness of the infrastructure.

6.8. Istanbul Byzantine Fault Tolerance (IBFT)

IBFT is simple byzantine fault-tolerant consensus algorithm mostly used for state machine replication in the quorum blockchain [166]. IBFT uses a partially synchronous communication model. It is deterministic, leader-based, and optimally resilient; it tolerates f faulty processes out of n, where $n \geq 3f + 1$. IBFT uses rounds. During each round, one of the processes acts as a leader and proposes a value. IBFT guarantees that all the correct processes reach a decision. A function leader selects the leader in a round. A recent modification in IBFT provides a higher throughput [167]. Two major changes between IBFT and enhanced IBFT are: (i) processes controlling block proposal verification are executed concurrently instead of running chronologically and (ii) decreased timeout between each phase.

6.9. Mixed Byzantine Fault Tolerance (MBFT)

MBFT is based on two layered BFT [168]. It uses sharding and layered technology. It introduces a random node selection mechanism and a credit mechanism to improve security and fault tolerance. Verifying nodes verify the transactions. The selection of verifying nodes is confirmed by other nodes in the network. The verifying nodes are classified into two layers: the low level consensus group (LCG) and the high-level consensus group (HCG). HCG obtains a miniblock from LCG after consensus, and checks for signature validation, hashing validation, and conflicts before further processing.

***PBFT and its variants, and IoT applicability*.** PBFT is the most popular consensus for IoTs. PBFT does not require high computing resources to reach consensus; therefore, it can be useful for constrained devices. Besides, PBFT has a fairly simple architecture to implement. This makes PBFT algorithm and its variants attractive for the rapid development of applied blockchain solutions. We summarize the applicability of PBFT and its variants for IoTs in Table 3.

7. DAG-based consensus

Blockchains show some disadvantages in terms of scalability and throughput. Besides, the concept of incentives to the miners also becomes illogical for micro-transactions. Dealing with these problems and advancing the decentralization with more benefits Directed Acyclic Graph (DAG) comes into existence [179]. Sompolinsky et al. introduce this concept to mainstream blockchains in the work of GHOST [180, 181]. The improved version of Sompolinsky et al.'s proposal is used in Ethereum [182]. However, the present status of DAG-based distributed ledger is in infancy stage. IOTA and Byteball are two major examples of cryptocurrencies using DAGs [183]. In this section, we review some recent developments on DAG-based blockchains and analyze their corresponding consensus methods.

7.1. Dagbase

Dagbase is a decentralized database platform that consists of three layers: Persistence Layer (PL), Image Layer (IL), and Application Layer (AL) [184]. PL stores immutable data immutably using a DAG-based consensus algorithm. Nodes can make deterministic decisions on consensus by analyzing the local DAG. It uses hashgraph consensus [185]. Dagbase considers each transaction as an event. The function *CreateEvent* creates the events where each of the events has a self-parent pointer and an other-parent pointer. This makes a DAG in Dagbase. The main purpose of this consensus is to share the newly created event amongst all the nodes by a *gossip-aboutgossip* protocol. Every non-faulty node can know the same DAG and reach consensus efficiently. The consensus is asynchronous BFT and avoids mining and incentive mechanisms.

7.2. Jointgraph

Jointgraph is based on byzantine fault-tolerance consensus algorithm [186]. It uses events to pack the transactions. The events are sent through a gossip protocol, which means anyone can send events to a random node. These events are validated by all the members. Jointgraph uses 2/3 of all the members as a threshold for consensus. Jointgraph uses a supervisor that monitors member behavior and improves consensus efficiency. The supervisory node collects the votes in the consensus process and determines the finality of the events. Every member has a copy of the Jointgraph, so one can calculate what vote the others would have sent to it. We show an example of Jointgraph consensus in Fig. 15. It considers three ordinary nodes (A, B, and C) and one supervisory node D. The dark circles represent the events with consensus and light circles are not in consensus. The dark circles are those events which are verified by at least three (the number of all nodes is four) members including the supervisory node. The confirmation time depends on the gossip frequency.

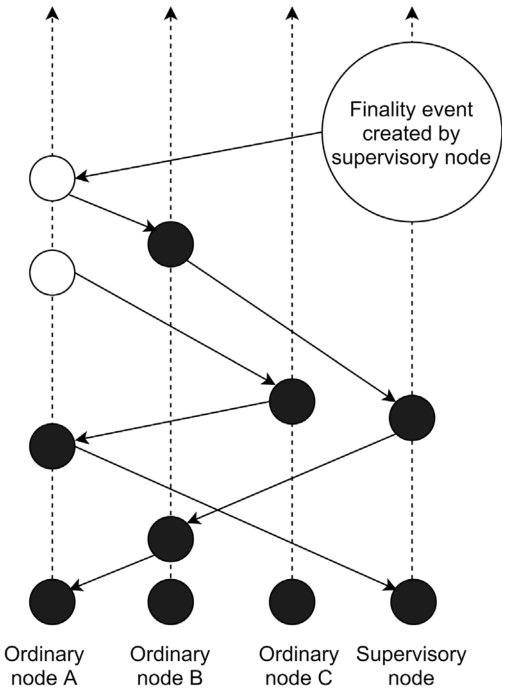


Fig. 15. Diagram representing blockchain structure.

7.3. BlockDAG

BlockDAG uses sorting and merges original blocks from a DAG structure [187]. It re-constructs a single-chain-based blockchain system. BlockDAG uses five phases: Block Generation (BG), Sorting Block (BS), Block Merging (BM), Consensus Implementation (CI), and Block Splits (BS). In the first phase, BG generates original blocks. When a transaction is generated, it is added to the block pool of the nearest blockchain node. The node validates all transactions in the block pool and bundles them up into a few blocks. These blocks are added to the system block pool that performs block validations. In the second phase, SB sorts all unvalidated blocks in the block pool. It uses a sorting algorithm to make a sequence for unvalidated blocks in the DAG structure. BlockDAG addresses the problems of double spending and consensus conflict problems. In BM phase, the merge occurs. These merged blocks finally proceed for a global consensus in CI phase. Finally, BS phase executes a process splitting the merged blocks into original states as well as locate blocks to on-premise BlockDAG structure.

7.4. UL-blockDAG

UL-blockDAG is a two-step consensus process for making the system robust against double-spending attacks [188]. In the first step, it uses a graph clustering algorithm based on spectral graph theory for separating the blocks created by the non-cooperating miners (attacker). The graph clustering executes two ways: maximizing the intra-cluster edge connections and minimizing the inter-cluster edge connections. UL-blockDAG uses Rayleigh–Ritz theorem for solving optimizing problem. It is an unsupervised learning based classification of the vertices of a graph into two classes. In the second step, it uses an ordering algorithm based on the topological ordering of the blockDAG using the references included in block header.

7.5. Dexon

DEXON consensus is based on PoP. In DEXON, every node is allowed to propose a block with equal probability [189]. Dexon uses Verifiable

Random Function (VRF) to decide on the issuer of a block. It reduces the communication cost encouraging more nodes to join the protocol. It uses the block lattice structure [190]. Dexon proposes the use of a fast byzantine agreement. It terminates in 6σ time, given σ is the upper bound of the network's gossip period. It generates on-chain unpredictable randomness on the fly as it achieves consensus. Once DEXON byzantine agreement confirms a block, a committee of nodes generates a threshold signature where the threshold is an unpredictable value. In this consensus, no single block proposer can determine the consensus timestamp of a proposed block. It achieves second-level latency instead of traditional minute-level latency showing its advantage. DEXON is highly decentralized and robust in practical deployment environments.

7.6. SPECTRE

SPECTRE is a new protocol for the consensus core of cryptocurrencies providing high throughput and fast confirmation time [190]. It provides high block creation rates. SPECTRE uses partial synchronous networks. SPECTRE generalizes Nakamoto's blockchain into a block DAG. It allows miners to create blocks concurrently by maintaining afull DAG of blocks. SPECTRE uses a voting algorithm regarding the order between each pair of blocks in the DAG. The voters are blocks (not miners); it interprets the vote of each block is algorithmically (and not provided interactively) according to its location within the DAG. In SPECTRE, the majority's aggregate vote becomes irreversible very fast; this majority vote provides a consistent set of transactions.

7.7. PHANTOM

PHANTOM is a PoW-based consensus protocol for a permissionless ledger. It generalizes Nakamoto's blockchain to a blockDAG [191]. It uses parameter k that controls the level of tolerance of the protocol to concurrently created blocks. Thus, it is able to provide higher throughput. PHANTOM uses the solution of an optimization problem over the blockDAG. It helps to distinguish between blocks mined properly by honest nodes and those created by non-cooperating nodes. This distinction provides a robust total order on the blockDAG. All the honest nodes agree upon this ordering. The implementation of PHANTOM requires solving an NP-hard problem. Authors use a greedy algorithm, called as *GHOSTDAG*, to provide the solution for PHANTOM.

DAG-based consensus and IoTs. DAG-based consensus protocols are attracting researchers for their benefits of data structures, high throughput, and scalability. In some cases, DAG-based consensus protocols avoid mining and incentives mechanisms. Therefore, the ease of configuration and less energy consumption make such consensus protocols applicable for IoTs. However, some issues and assumptions are necessary to enhance these DAG-based consensus to be suitable for IoT heterogeneity. We summarize the existing DAG-based consensus protocols with their advantages, disadvantages, and IoT applicability in Table 4.

8. Analysis

In this section, we show an analysis of the consensus protocols described in previous sections. It has three subsections. Section 8.1 provides categorization of the consensus protocols based on the application domain. Section 8.2 shows the categorization of the consensus protocols based on the blockchain types. Section 8.3 compares the existing surveys of consensus with the present survey.

8.1. Application-based categorization

Blockchain applications have expanded in various domains. Apart from cryptocurrencies consensus are also usable for various applications in the domain of finance, security, supply chain management, database management and governance. Moreover, there are lots of consensus protocols exist which work as deployment platform for blockchain or distributed ledger applications. Table 5 enlists various applications in the purview of consensus protocols.

8.2. Consensus and blockchain types

These consensus approaches are used for four different blockchain types: public, private, consortium, and hybrid. Though there is no hard rule to say a public blockchain is applicable only for public setup only; the configuration of the consensus can be changed to suit a particular type of blockchain.

- *Public blockchain*: It is non-restrictive and permissionless. Any user with internet access can sign on to a blockchain platform to become an authorized node. This user can access transaction history and update themselves accordingly. Transactions are anonymous but transparent. Examples: Bitcoin, Ethereum, Litecoin, NEO.
- *Private blockchain*: A private blockchain works in a restrictive environment, i.e., a closed network. It is a permissioned blockchain that is under the control of an entity. Thus, it have a centralization tendency. Private blockchains are good for using at a privately-held company or organization that wants to use it for internal use-cases. Example: Multichain, Hyperledger Fabric, Hyperledger Sawtooth, Corda.
- *Consortium blockchain*: A consortium blockchain (also known as Federated blockchains) solves organizations' needs where there is a need for both public and private blockchain features. In a consortium blockchain, some aspects of the organizations are made public, while others remain private. It is a kind of hybrid orientation. A consortium blockchain is managed by more than one organization. Example: Marco Polo, Energy Web Foundation, IBM Food Trust.
- *Hybrid blockchain*: Hybrid blockchain may seem to follow a consortium blockchain, but it is different. Hybrid blockchain is a combination of a private and public blockchain. It has use-cases in an organization that neither wants to deploy a private blockchain nor public blockchain directly. Typically, transactions and records in a hybrid blockchain are not made public but can be verified when needed, such as by allowing access through a smart contract. Example of Hybrid Blockchain: Dragonchain, XinFin's Hybrid blockchain

Consensus algorithms are also classified in two parts: proof-based and leader-based. In proof-based consensus, the members in the network need to prove that some activity or condition has been fulfill by a member that gives them the right to add block and obtain rewards. In vote-based approach, the members of the network tend to vote for each decision and the majority, or the most probable decision is accepted by all. In some cases, a leader is selected who will represent the group of members and perform activity on their behalf. The key difference observed is that proof based approach is usually used for public blockchains, but the vote-based approach is usually used for private or consortium blockchain where the identity of the members are known. Table 6 shows the classification of consensus protocols based on blockchain type and Table 7 shows the classification of consensus protocols based on consensus type.

8.3. Comparison with existing surveys

We provide a summary of the existing surveys on consensus in Table 8. The present survey of consensus protocols differs from the existing surveys in the following points.

- *Classification of consensus*: We classify the consensus in various categories. The very first segregation is based on structure: blockchain-based and DAG-based. In blockchain-based consensus, we further show the consensus protocols based on their genesis theory: Proof-of-X, Paxos, RAFT, and Byzantine. Overall, 58 consensus protocols are reviewed. To the best of our knowledge, our survey is the most comprehensive and updated survey in the direction of consensus protocols.

Table 4
Summarization of DAG-based consensus protocols.

Name of Algorithm	Ref.	Advantage	Disadvantage	IoT applicability
Dagbase	[184]	Faster, less energy requirement, no miner and incentives	Dagbase uses hashgraph which inherits some problems such as control mechanism, inability to support turing-complete smart contract, and possible corruption of hashgraphs	Optimized event sharing and classification of events are required for IoTs as such infrastructures are heterogeneous and dynamic in terms of network participation
Jointgraph	[186]	Robust against double spending attack, improved throughput as compared to hashgraph	A single supervisory node can be a target of attack	The selection of supervisory node in IoT backbone needs separate consideration and strong security
BlockDAG	[187]	Scalable, high throughput, robustness	parallel processing needs assurance of transaction records once, latency increases with merge sort, splitting process can be prone to attack	The merging and sorting of blocks can be a hard task in heterogeneous IoT applications, efficient sorting merging operations required such as for Hadoop clusters
UL-blockDAG	[188]	Higher block creation rate and transaction throughput, robustness	The complexity increases with increasing number of nodes	Due to the use of unsupervised classification, the accuracy is less for large heterogeneous IoT networks
Dexon	[189]	Low latency, higher throughput, reduced cost, transaction ordering fairness, scalable, unpredictable randomness, secure transaction finality, useful for resource constrained environments	Not identified yet	Features of Dexon are applicable for IoT paradigm; needs further study with different applications
SPECTRE	[190]	High throughput, scalable, high block creation rate	delay of transactions for visibly double-spending transactions should be explored further	As IoT intrinsically possess security threats, the malicious voting may lead to overall consensus failure, security of the consensus voting must be ensured properly
PHANTOM	[191]	High security threshold, improved throughput	Security analysis required, confirmation times of GHOSTDAG	The selection of greedy algorithm in IoT is a challenge due to compatibility of the heterogeneous devices

- *Analysis*: We analyze the individual protocols based on their pros and cons. We also try to link the variations of the consensus protocols from their genesis source. We also include a contemporary challenges in these protocols which need to be addressed in future; thus providing some updated open research problems for the readers.
- *DAG-based consensus*: None of the existing survey provides information about DAG-based consensus; however, such consensus protocols are attracting researcher communities and thus, they are also the point of interest in our survey.

9. Open research problem

Each sections of the consensus protocols summarize the pros and cons for the corresponding protocol. All the cons of the protocols lead to some research problems to solve. In this section, we summarize all these problems and group them as the overall understanding of the blockchains and consensus directions.

- *Energy efficiency:* PoW-based approaches show increased resource consumption. Therefore, such consensus protocols are not suitable for IoTs and other resource constrained environments. However, the popularity of PoW-based approaches are optimum. Therefore, the enhancements to such consensus protocols must be followed in future.
- *Safety:* By safety we mean that the members of the network should not decide on different values individually. Even if there are some faulty members, maximum members should decide on the same value. Safety also means that no matter what happens nothing bad will happen.
- *Security:* The approach should be resistant against various attacks. Some attacks are 51% attack, double spending problem, balance attack, sybil attack, eclipse attack, finney attack, etc. Maximum developing consensus protocols avoid the security and concentrate more on finality, throughput, latency, and confirmation times. Therefore, the security-scalability-decentralization trilemma still persists. Very few consensus protocols work on this and thus, it still opens a promising research direction.
- *Scalability:* When the number of members or nodes in the network increases, the system should perform in similar manner as before. Thus, the system should be able to well under an increased load. Though in recent times, the DAG-based blockchains are solving this issue, the infancy stage of these protocols can be enhanced towards maturity with more improvement.
- *Finality:* Finality says that once a block containing transactions is added to the chain, it will not be revoked. Two types of finality exist: probabilistic finality and absolute finality. Chain-based consensus algorithms provide probabilistic finality while the consensus approaches like byzantine-based algorithms provide absolute finality. For a blockchain system, the latter type is better than the former one and thus, more consideration required.
- *Liveness:* Liveness tells about the nodes will reach a decision. The chosen consensus algorithm should reach a final decision and preferably within a prescribed period. It will be of no use if no decision is reached after consensus. Therefore, any development of consensus must check for finality time requirements.
- *Decentralization:* It is a key feature of the blockchain systems. It is observed that public blockchains offer more decentralization than the private blockchains. To maintain security, the decentralization should not be abandoned completely at all. As the consensus

Table 5
Categorization of consensus protocols in different applications.

No.	Application Domain	Technology	Consensus Algorithm
1.	Cryptocurrency	Bitcoin [36]	Proof of Work
		Ethereum [192]	Proof of Work
		NXT [51]	Proof of Stake
		SlimCoin [57]	Proof of Burn
		PeerCoin [52]	Proof of Stake
		BurstCoin [55]	Proof of Space (Capacity)
		SpaceMint [193]	Proof of Space (Capacity)
		ReddCoin [63]	Proof of Stake Velocity
		NEM [173]	Proof of Importance
		PoP Coins [108]	Proof of Personhood
		Mobilecoin [149]	Stellar Consensus Protocol
		Ripple [176]	XRP Ledger Consensus Protocol
		Neo [177]	Delegated BFT
		Dash [194]	Proof of Stake
		Litecoin [195]	Proof of Work
		EOS [196]	Delegated Proof of Stake
		Decred [71]	Proof of Activity
		Vet (VeChain) [101]	Proof of Authority
		Lisk [79]	Delegated Proof of Stake
		Ada [48]	Ouroboros
		IOTA [179]	FPC
2.	Database management	Apache Casandra [134]	PAXOS Algorithm
		Distributed Systems Google Chubby Lock Service [136]	PAXOS Algorithm
		Google's Megastore [145]	PAXOS Algorithm
		Cockroach DB [197]	RAFT Algorithm
		Apache Kudu [134]	RAFT Algorithm
3.	Payment system and finance	Counterparty [198]	Proof of Burn
		BitShare [78]	Delegated Proof of Stake
		Chia [59]	Proof of Space and Time
		Kovan Testnet [100]	Proof of Authority
		Algorand [199]	Proof of Stake (variant)
		Ripple [196]	XRP Ledger Consensus Algorithm
		Stellar [148]	Stellar Consensus Protocol
		Cardano [48]	Ouroboros
4.	Supply chain management	VieChain [101]	Proof of Authority
5.	Governance	Decreed [71]	Proof of Activity
		Algorand Protocol [199]	Proof of Stake (variant)
		PoP Coin [108]	Proof of Personhood
6.	Platform for development	Ethereum [192]	Proof of Work
		TenderMint [172]	PBFT Algorithm
		Hyperledger [171]	PBFT Algorithm
		Komodo Project [75]	Delayed Proof of Work
		Lisk [79]	Delegated Proof of Stake
		NEM [89]	Proof of Importance
		Microsoft Azure [96]	Proof of Authority
		GoChain [200]	Proof of Reputation
		EOS [196]	Delegated Proof of Stake
		Cardano [48]	Ouroboros
7.	Enterprise use cases	Sawtooth [84]	Proof of Elapsed Time
		GoChain [107]	Proof of Reputation
8.	Additional security layer	PeerCoin [52]	Proof of Stake
		BitShares [78]	Delegated Proof of Stake
		Komodo Project [75]	Delayed Proof of Work

Table 6
Categorization of consensus protocols based on blockchain type.

Blockchain type	Applicable consensus
Public	PoW, PoS, PoB, PoSC, PoSV, PoA, DPoW, DPoS, PoET PoST, PoI, PoWt, PoR, PoP, PoL, PoFL, PoInd, PoX, PoFPoW, PoLID, PPoW, PoPr, PoSin, PoRel, PoWP, PoApp, SCP, Byzantine agreement based consensus, dBFT, FPC, IBFT, DAG-based consensus
Private	DPoW, PoAu, PoWt, PoLu, PoM, PoWP, Paxos and its variants, RAFT, PBFT, XRP ledger consensus, dBFT, Multi-signature BFT
Consortium	PoET, PoR, PoV, Reputation based BFT, MBFT

Table 7
Categorization of consensus protocols based on consensus type.

Consensus type	Applicable consensus
Proof-based	PoW, PoB, PoSC, PoSV, PoA, DPoW, DPoS, PoET PoST, PoI, PoWt, PoR, PoP, PoL, PoFL, PoInd, PoX, PoFPoW, PoLID, PPoW, PoPr, PoRel, PoWP, PoApp
Vote-based	Paxos and its variants, SCP, Ouroborous (based on PoS), BFT and its variants, RAFT, DPoS, PoS, PoWt, PoV, PoSin

are the integral part of blockchains, the new developments of consensus must experimentally validate about their decentralization and security.

- *Correctness and progress:* The decision taken by the system should be correct. If a decision that is taken within a short time is not correct, then it will be of no use. There should not be any deadlock in the process. If any deadlock occurs, there must be a mechanism to overcome that.

Table 8
Features of existing surveys.

Reference	Year	Features discussed
[201]	2017	Comparative analysis of PoW, PoS, DPoS, PBFT and RAFT along with the applications.
[202]	2017	Comparative analysis of PoW, PoS, BFT, FBA and PoET.
[203]	2018	Comparative analysis of proof-based vs vote-based mechanisms, PoW-based and PoS-based, hybrid consensus.
[204]	2018	Design factors of consensus protocols, incentive mechanisms.
[205]	2018	Comparative analysis of PoW-based, PoS- based, Byzantine agreement based, VRF-based methods, sharding-based methods, RAFT and Tangle.
[206]	2018	Comparative analysis of PoW, PoS, PBFT and zero knowledge proofs; incentive models.
[207]	2018	Comparative analysis of nine consensus algorithms based on performance and security.
[208]	2019	Comparative study of common consensus algorithms.
[209]	2019	Categorization of proof-based and vote based consensus; comparative analysis on incentive, performance, exposure and others.
[210]	2019	Comparative study of PoW, PoS, PoB, Paxos, Raft and BFT along with its variants.
[211]	2019	Evolution of consensus algorithms with 15 variants.
[212]	2019	Comparative analysis of PoW, PoS, PBFT, PoB, DPoS, LPoS, POA, PoI and Proof of Weight.
[213]	2019	Comparison of permissioned consensus algorithms and permission-less consensus algorithms.
[214]	2019	Taxonomy of consensus algorithms in hardware based, vote based and stake based categories.
[215]	2020	Comparative analysis of 23 consensus algorithms for block proposal, block Validation, information preposition, block finalization and incentive mechanism.
[216]	2020	Timeline approach of consensus evolution; comparative analysis of 25 consensus algorithms.
[217]	2020	Comparative analysis of PBFT, PoA, IBFT (Istanbul Fault Tolerance) and CFT (Crash Fault Tolerance).
[218]	2020	Comparative study of probabilistic consensus algorithms and deterministic consensus algorithm.
[219]	2020	Distributed consensus mechanism; analysis of PoW and PoS.
[220]	2020	Pros and cons of 24 consensus algorithms.
[221]	2020	Categorization of permissioned and permissioned consensus algorithms; blockchain implementation in the form of Hyperledger fabric, Corda, Quorum and Ethereum.
[222]	2021	28 consensus protocols are discussed, a four-category classification framework based on origin, design, performance, and security; application classification of 28 protocols.

- *Storage Requirement:* Almost all the blockchain networks require the nodes or members to download the state of the network (full or light) which requires a certain amount of storage space. Moreover, the verifiability of the consensus decision also requires space. Thus, this space requirement should be optimized for the consensus developments to avoid any storage overhead.
- *Application:* The choice of consensus algorithm depends upon the purpose for which the overlying blockchain is developed, therefore various applications' acceptability need to be explored more. In the literature, we have reviewed overall 58 consensus protocols; however, very few are ready for application. Therefore, the selection of consensus protocols must be verified. We should also seek further for a benchmark test or an benchmark framework for consensus protocols to follow which is lacking at the moment.
- *Anonymity:* While maintaining transparency is a desirable property for a better blockchain system, anonymity must not be sacrificed. Therefore, some algorithms propose that users use public key, private key and public address instead of presenting their real information. There is also a 12-word sequence of letters (also called seed phrase) used by few platforms that link it with a specific account detail. This also maintains anonymity and security.
- *Understandability:* There are various algorithms such as Paxos which are very hard to understand. Though there are different variations of Paxos available which try to generalize the understandability; the basic establishment is not so strong for Paxos like protocols. Thus, these protocols should be revised by the researchers.
- *DAG-based consensus:* DAG-based blockchains are showing their advantages of scalability, higher throughput, feeless transactions. Thus, they are more in interest of the researchers. Therefore, the consensus protocols in such blockchains must be enhanced to support existing blockchain structures.

10. Conclusion

Blockchain has its potential to prove its significance of usage in various applications. The above discussion of existing consensus approaches and their pros and cons give a clear impression that blockchain has spread its varsity with its promising features in present time of IoTs. We have reviewed total 58 consensus protocols available till date which more than the existing surveys. The technical attributes and detailed functions make the readers comfortable to understand the basic of the consensus. This review work is beneficial for future blockchain researchers to design a protocol or to explore the applications as per the requirements.

CRediT authorship contribution statement

Arshdeep Singh: Data curation, Writing – original draft. **Gulshan Kumar:** Data curation, Writing – original draft. **Rahul Saha:** Data curation, Writing – original draft. **Mauro Conti:** Data Interpretation, Supervision. **Mamoun Alazab:** Data Analysis, Manuscript correction. **Reji Thomas:** Data Analysis, Data Interpretation.

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] The paxos algorithm - YouTube, 2020, Accessed:22-Sep-2020.

[2] Comparison - centralized, decentralized and distributed systems - GeeksforGeeks, 2020, [Online]. Available: Accessed:03-Sep-2020.

[3] A.T. Sherman, F. Javani, H. Zhang, E. Golaszewski, On the origins and variations of blockchain technologies, IEEE Secur. Priv. 17 (1) (2019) 72–77.

[4] A. Srivastava, P. Bhattacharya, A. Singh, A. Mathur, A systematic review on evolution of blockchain generations, Inf. Technol. Electr. Eng. J. 7 (6) (2018) 1–8.

[5] M. Li, D. Hu, C. Lal, M. Conti, Z. Zhang, Blockchain-enabled secure energy trading with verifiable fairness in industrial internet of things, IEEE Trans. Ind. Inf. 16 (10) (2020) 6564–6574.

[6] M. Conti, M. Hassan, C. Lal, Blockauth: BlockChain based distributed producer authentication in ICN, Comput. Netw. 164 (2019).

[7] J.C. Shah, M. Bhagwat, D. Patel, M. Conti, Crypto-wills: Transferring digital assets by maintaining wills on the blockchain, in: J. Bansal, M. Gupta, H. Sharma, B. Agarwal (Eds.), Communication and Intelligent Systems. ICCIS 2019, in: Lecture Notes in Networks and Systems, vol. 120, Springer, Singapore.

[8] M. Li, C. Lal, M. Conti, D. Hu, Lechain: A blockchain-based lawful evidence management scheme for digital forensics, Future Gener. Comput. Syst. 115 (2021) 406–420.

[9] Mohammad Shahriar Rahman, Abdullah Al Omar, Md. Zakirul Alam Bhuiyan, Anirban Basu, Shinsaku Kiyomoto, Guojun Wang, Accountable cross-border data sharing using blockchain under relaxed trust assumption, IEEE Trans. Engineering Management 67 (4) (2020) 1476–1486.

[10] Lin Liu, Wei-Tek Tsai, Md. Zakirul Alam Bhuiyan, Hao Peng, Mingsheng Liu, Blockchain-enabled fraud discovery through abnormal smart contract detection on ethereum, Future Gener. Comput. Syst. 128 (2022) 158–166.

[11] A.S. Bruyn, Blockchain an introduction, 2017.

[12] ConsensUS | meaning in the cambridge english dictionary, 2020, [Online]. Available: Accessed:02-Sep-2020.

[13] J. Ousterhout, Designing for understandability: The raft consensus algorithm - YouTube, CS @ illinois distinguished lecture series, 2016, [Online]. Available: Accessed:23-Jul-2020.

[14] S. Zhang, J.H. Lee, Analysis of the main consensus protocols of blockchain, ICT Express (2019) 1–5, no. xxxx.

[15] Pros and cons of different blockchain consensus protocols, 2020, [Online]. Available: Accessed:02-Sep-2020.

[16] Private blockchain consensus mechanisms - dulguun batmunkh - medium, 2020, [Online]. Available: Accessed:02-July-2020.

[17] D. Yaga, D. Yaga, Nist.Ir.8202.

[18] Rabeya Bosri, Mohammad Shahriar Rahman, Md. Zakirul Alam Bhuiyan, Abdullah Al Omar, Integrating blockchain with artificial intelligence for privacy-preserving recommender systems, IEEE Trans. Netw. Sci. Eng. 8 (2) (2021) 1009–1018.

[19] Lin Liu, Wei-Tek Tsai, Md. Zakirul Alam Bhuiyan, Dong Yang, Automatic blockchain whitepapers analysis via heterogeneous graph neural network, J. Parallel Distributed Comput. 145 (2020) 1–12.

[20] Youliang Tian, Ta Li, Jinbo Xiong, Md. Zakirul Alam Bhuiyan, Jianfeng Ma, Changgen Peng, A blockchain-based machine learning framework for edge services in iIoT, IEEE Trans. Ind. Informatics 18 (3) (2022) 1918–1929, Regards.

[21] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, in: Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, No. June, 2017, pp. 557–564.

[22] P. Tasca, C.J. Tessone, A taxonomy of blockchain technologies: Principles of identification and classification, 5980, 2019, pp. 1–39.

[23] S. Notions, S. Al-kuwari, J.H. Davenport, R.J. Bradford, Cryptographic hash functions: Recent design trends and, eprint.iacr.org, 2011, pp. 1–36.

[24] K. Zile, R. Strazdina, Blockchain use cases and their feasibility, Appl. Comput. Syst. 23 (1) (2018) 12–20.

[25] T. Primer, Blockchain, 2018, pp. 0–38, no. July.

[26] Blockchain nodes: How they work (all types explained) - nodes.com, 2020, [Online]. Available: Accessed:09-Sep-2020.

[27] V. Gramoli, From blockchain consensus back to Byzantine consensus, Futur. Gener. Comput. Syst. (2017).

[28] What is a peer to peer network? Blockchain P2P networks explained - YouTube, 2020, [Online]. Available: Accessed:02-Sep-2020.

[29] H.F. Atlam, A. Alenezi, M.O. Alassafi, G.B. Wills, Blockchain with internet of things: Benefits, challenges, and future directions, Int. J. Intell. Syst. Appl. 10 (6) (2018) 40–48.

[30] Blockchain architecture explained: How it works and how to build it | mlsdev, 2020, [Online]. Available: Accessed:03-Sep-2020.

[31] K. Sultan, U. Ruhi, R. Lakhani, Conceptualizing blockchains: Characteristics & applications, in: Proc. 11th IADIS Int. Conf. Inf. Syst. 2018, IS 2018, 2018, pp. 49–57.

[32] Fat protocols | union square ventures, 2020, [Online]. Available: Accessed:04-Feb-2020.

[33] The blockchain technology stack - arun devan - medium, 2020, [Online]. Available: Accessed:24-Sep-2020.

[34] M. Jakobsson, A. Juels, Proofs of work and bread pudding protocols(extended abstract), Secur. Inf. Networks (1999) 258–272.

[35] C. Dwork, M. Naor, Prciing via processing or combatting junk mail, in: Brickell E.F. Adv. Cryptol. — CRYPTO' 92. CRYPTO 1992, in: Lect. Notes Comput. Sci., vol. 740, Springer, Berlin, Heidelb, 1993.

[36] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2016, pp. 1–9.

[37] J. Garcia-Alfaro, et al., Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8872, 2015, pp. 3–16.

[38] A quick introduction to bitcoin | get started | bitcoin.com, 2020, [Online]. Available: Accessed:05-Feb-2020.

[39] S.S. Hazari, Q.H. Mahmoud, A parallel proof of work to improve transaction speed and scalability in blockchain systems, in: 2019 IEEE 9th Annu. Comput. Commun. Work. Conf., 2019, pp. 916–921.

[40] H. Vranken, Sustainability of bitcoin and blockchains, Curr. Opin. Environ. Sustain. 28 (2017) 1–9.

[41] S. Grewal, Komodo's delayed proof of work (dpow) security, explained, 2020, [Online]. Available: Accessed:04-Sep-2020.

[42] K. Li, H. Li, H. Hou, K. Li, Y. Chen, Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain, 2017.

[43] bitcointalk.org, Proof of stake instead of proof of work, 2011, Accessed:03-Sep-2020.

[44] W. Yan, M. Maung, Formal analysis of a proof-of-stake blockchain, in: 2018 23rd Int. Conf. Eng. Complex Comput. Syst, 2018, pp. 197–200.

[45] Proof of burn explained | binance academy, 2020, [Online]. Available: Accessed: 03-Sep-2020.

[46] C. Elisabetta, Z. Baltico, D. Catalano, D. Fiore, R. Gay, Advances in cryptology, in: CRYPTO 2017-37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August (2017) 20-24, Proceedings, Part III, 10403, 2017, 639554.

[47] Iohk | cardano whiteboard; overview with charles hoskinson - YouTube, 2020, [Online]. Available: Accessed:10-Feb-2020.

[48] Ouroboros - cardano, 2020, [Online]. Available: Accessed:10-Feb-2020.

[49] Binance academy, what is proof of stake? - YouTube, 2018, [Online]. Available: Accessed:03-Sep-2020.

[50] BitFury Group, Proof of stake versus proof of work, BitFury Gr. 2015 (2015) 1–26.

[51] What is nxt? 2019 beginner's guide on NXT cryptocurrency, 2020, Accessed: 01-Sep-2020.

[52] Peercoin — The pioneer of proof of stake, 2020, Accessed:01-July-2020.

[53] T. Shuliar, N. Goldsmit, Proof of value alienation (PoVA) - A concept of a cryptocurrency issuance protocol, 2019.

[54] Slimcoin a peer-to-peer crypto-currency with proof-of-burn ;mining without powerful hardware, 2014.

[55] What is counterparty (XCP)? | beginner's guide - CoinCentral, 2020, [Online]. Available: Accessed:03-Sep-2020.

[56] Slimcoin's advantages, 2020, [Online]. Available: Accessed:03-Sep-2020.

[57] About slimcoin, 2020, [Online]. Available: Accessed:03-Sep-2020.

[58] S. Dziembowski, S. Faust, V. Kolmogorov, K. Pietrzak, Proofs of Space, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9216(616160), 2015, pp. 585–605.

[59] Home - chia network, 2020, [Online]. Available: Accessed:06-Sep-2020.

[60] G. Ateniese, I. Bonacina, A. Faonio, N. Galesi, Proofs of Space: When Space Is of the Essence, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8642, 2014, pp. 538–557.

[61] Blockchain – burstcoin, 2020, [Online]. Available: Accessed:06-Sep-2020.

[62] Home - burst wiki, 2020, [Online]. Available: Accessed:06-Sep-2020.

[63] Reddcoin.com – the social currency, 2020, [Online]. Available: Accessed:07-Sep-2020.

[64] L. Ren, Proof of stake velocity: Building the social currency of the digital age, 2014, pp. 1–13.

[65] Reddcoin announces rollout of new enhanced proof of stake velocity protocol to bolster network growth and stability - MarketWatch, 2020, [Online]. Available: Accessed:05-Feb-2020.

[66] Proof of activity proposal, 2020, [Online]. Available: Accessed:08-Sep-2020.

[67] I. Bentov, C. Lee, A. Mizrahi, 4-0 proof of activity: PoW+PoS, (240258), 2013, pp. 1–19.

[68] Proof-of-activity (PoA) - wiki | golden, 2020, [Online]. Available: https://golden.com/wiki/Proof-of-activity_(PoA). [Accessed: 08-Sep-2020].

[69] Proof of activity explained: A hybrid consensus algorithm - coin bureau, 2020, [Online]. Available: Accessed:05-Feb-2020.66.

[70] Overview - decred documentation, 2020, [Online]. Available: Accessed:08-Sep-2020.

[71] Decred - autonomous digital currency, 2020, [Online]. Available: Accessed:03-Sep-2020.

[72] Delayed proof of work explained | binance academy, 2020, [Online]. Available: Accessed:04-Sep-2020.

[73] Delayed proof of work (dpow) whitepaper. SuperNETorg/komodo wiki. GitHub, 2020, [Online]. Available: Accessed:04-Sep-2020.

[74] Delayed proof of work: the multi-blockchain consensus algorithm, 2020, [Online]. Available: Accessed:07-Feb-2020.

[75] Security: Delayed proof of work (dpow) - komodo, 2020, [Online]. Available: Accessed:04-Sep-2020.

[76] F.A.N. Yang, et al., Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism, IEEE Access 7 (2019) 118541–118555.

[77] What is delegated proof of stake (dpos) - explained for beginners - YouTube, 2019, [Online]. Available: Accessed:03-Sep-2020.

[78] Delegated proof-of-stake consensus | BitShares blockchain, 2020, [Online]. Available: Accessed:03-Sep-2020.

[79] Lisk SDK overview: Documentation, 2020, [Online]. Available: Accessed:04-Sep-2020.

[80] P. Brain, S.A. Recovery, Proof of brain: Smart and social tokens, 2017, pp. 7–13.

[81] B.B. Foundation, Bitshares - technical white paper, 2018.

[82] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, W. Shi, On Security Analysis of Proof-of-Elapsed-Time (PoET), in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), LNCS, vol. 10616, 2019, pp. 282–297, (2017).

[83] Understanding hyperledger sawtooth — Proof of elapsed time, 2020, [Online]. Available: Accessed:04-Sep-2020.

[84] Introduction — Sawtooth latest documentation, 2020, [Online]. Available: Accessed:04-Sep-2020.

[85] Poet 1.0 specification — Sawtooth v1.0.5 documentation, 2020, [Online]. Available: Accessed:04-Sep-2020.

[86] Chia network - YouTube, 2020, [Online]. Available: Accessed:06-Sep-2020.

[87] B. Cohen, K. Pietrzak, The chia network blockchain, 1, 2019, pp. 1–44.

[88] Chia network proves reality of consensus-based on proof of space; announces proof of space competition with $100k in total prize money, 2020, [Online]. Available: Accessed:06-Sep-2020.

[89] What is POI | NEM documentation, 2020, [Online]. Available: Accessed:06-Sep-2020.

[90] Introduction to NEM (XEM): The proof-of-importance coin | CryptoSlate, 2020, [Online]. Available: Accessed:06-Sep-2020.

[91] NEM, NEM white paper, 2018.

[92] NEM lightwallet released, 2020, [Online]. Available: Accessed:06-Sep-2020.

[93] All about NEM (XEM), the harvested cryptocurrency, 2020, [Online]. Available: Accessed:04-Sep-2020.

[94] FAQ: What is PoI? (proof of importance) - NEM Singapore - medium, 2020, [Online]. Available: Accessed:06-Sep-2020.

[95] Proof of authority explained | binance academy, 2020, [Online]. Available: Accessed:07-Sep-2020.

[96] Ethereum proof-of-authority consortium - azure | microsoft docs, 2020, [Online]. Available: Accessed:07-Sep-2020.

[97] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, Pbft vs proof-of-authority: Applying the CAP theorem to permissioned blockchain, in: CEUR Workshop Proc., Vol. 2058, 2018, pp. 1–11.

[98] What is azure—Microsoft cloud services | microsoft azure, 2020, [Online]. Available: Accessed:04-Sep-2020.

[99] What is proof of authority consensus? (PoA) staking your identity, 2020, [Online]. Available: Accessed:07-Sep-2020.

[100] Announcing kovan — A stable ethereum public testnet, 2020, [Online]. Available: Accessed:04-Sep-2020.

[101] A beginner's guide to VeChain thor (VET) | UseTheBitcoin, 2020, [Online]. Available: Accessed:04-Sep-2020.

[102] What is proof of weight? | CoinCodex, 2020, [Online]. Available: Accessed:16-Sep-2020.

[103] A hitchhiker's guide to consensus algorithms - by, 2020, [Online]. Available: Accessed:16-Sep-2020.

[104] Consensus algorithms: The root of the blockchain technology, 2020, [Online]. Available: Accessed:16-Sep-2020.

[105] S. Z. B, Y. Rong, Y. Zheng, H. Cheng, J. Huang, Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-To-Peer Network, Vol. 1, Springer International Publishing, 2018.

[106] The future of blockchain: Proof of reputation - GoChain - medium, 2020, [Online]. Available: Accessed:17-Sep-2020.

[107] Proof of reputation - GoChain - medium, 2020, [Online]. Available: Accessed:17-Sep-2020.

[108] M. Borge, E. Kokoris-kogias, P. Jovanovic, L. Gasser, N. Gailly, B. Ford, 2017 European Proof-of-personhood: Redemocratizing permissionless cryptocurrencies, 2017.

[109] Personhood.online, 2020, [Online]. Available: Accessed:04-Sep-2020.

[110] F. Bravo-Marquez, S. Reeves, M. Ugarte, Proof-of-learning: A blockchain consensus mechanism based on machine learning competitions, 2019, pp. 119–124, no. January.

[111] Xidi Qu, Shengling Wang, Qin Hu, Xiuzhen Cheng, Proof of federated learning: A novel energy-recycling consensus algorithm, IEEE Trans. Parallel Distrib. Syst. 32 (8) (2021) 2074–2085.

[112] Proof-of-individuality, 2020, [Online]. Available: Accessed:26-Sep-2020.

[113] Proof-of-individuality, 2020, [Online]. Available: Accessed:26-Sep-2020.

[114] M. Milutinovic, W. He, H. Wu, M. Kanwal, Proof of luck: An efficient blockchain consensus protocol, in: SysTEX 2016-1st Work. Syst. Softw. Trust. Exec. Coloca. with ACM/IFIP/USENIX Middlew. 2016, 2016, pp. 2–7.

[115] A. Shoker, Sustainable Blockchain through Proof of eXercise.

[116] R. Nakahara, Proposal of fair proof-of-work system based on rating of user ' s computing power, in: 2018 IEEE 7th Glob. Conf. Consum. Electron., pp. 746–748.

[117] J. Kim, A study on an energy-effective and secure consensus algorithm for private blockchain systems (pom: Proof of majority), in: 2018 Int. Conf. Inf. Commun. Technol. Converg, 2018, pp. 932–935.

[118] T. Ogawa, H. Kima, N. Miyaho, Proposal of proof-of-lucky-ID (PoL) to solve the problems of PoW and PoS, in: 2018 IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, 2018, pp. 1212–1218.

[119] C. Ehmke, F. Wessling, C.M. Friedrich, Proof-of-property – a lightweight and scalable blockchain protocol, in: 2018 IEEE/ACM 1st Int. Work. Emerg. Trends Softw. Eng. Blockchain, 2018, pp. 48–51, no. January.

[120] M.U. Zaman, M. Min, Proof of sincerity: A new lightweight consensus approach for mobile blockchains, in: 2019 16th IEEE Annu. Consum. Commun. Netw. Conf. (2018), 2019, pp. 1–4.

[121] Por/BasicConsensus.md at master.ninanoo/por.GitHub, 2020, [Online]. Available: Accessed:27-Sep-2020.

[122] E. Pournaras, Proof of witness presence: Blockchain consensus for augmented democracy in smart cities, 2019.

[123] S. Takahashi, Proof-of-approval: A distributed consensus protocol for blockchains, 2018, pp. 1–21.

[124] Proof of burn | consensus through coin destruction, 2020, [Online]. Available: Accessed:03-Sep-2020.

[125] D.M. Eres, The stellar consensus protocol: A federated model for internet-level consensus, 2016.

[126] How the stellar consensus protocol (federated Byzantine agreement) works - YouTube, 2020.

[127] Consensusday 1 // stellar consensus protocol - david mazieres - YouTube, 2020.

[128] L. Lamport, Paxos made simple, 2001.

[129] Roberto De Prisco, Butler Lampson, Nancy Lynch, Revisiting the paxos algorithm, Theoret. Comput. Sci. 243 (1–2) (2000) 35–91.

[130] https://github.com/dgryski/awesome-consensus.

[131] Paxos agreement - computerphile - YouTube, 2020, [Online]. Available: Accessed:22-Jul-2020.

[132] S. Chakraborty, Lecture 18: Permissioned blockchain – v (practical Byzantine fault tolerance) - YouTube, 2020, [Online]. Available: Accessed:28-Jul-2020.

[133] Apache cassandra: The truth behind tunable consistency, lightweight transactions ; secondary indexes - the distributed SQL blog, 2020, [Online]. Available: Accessed:15-Sep-2020.

[134] Apache cassandra, 2020, [Online]. Available: Accessed:31-Sep-2020.

[135] J. Baker, et al., Megastore: Providing scalable, highly available storage for interactive services, in: CIDR 2011-5th Bienn. Conf. Innov. Data Syst. Res. Conf. Proc., 2011, pp. 223–234.

[136] M. Burrows, The Chubby lock service for loosely-coupled distributed systems.

[137] Saksham Chand, Yanhong A. Liu, Scott D. Stoller, Formal Verification of Multi-Paxos for Distributed Consensus, available at: https://arxiv.org/pdf/1606.01387.pdf.

[138] Leslie Lamport, Mike Massa, Cheap paxos, in: International Conference on Dependable Systems and Networks (DSN 2004), 2004, pp. 307–314.

[139] Leslie Lamport, Fast Paxos, available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-112.pdf.

[140] Miguel Castro, Barbara Liskov, Practical Byzantine fault tolerance, proceedings of the third symposium on operating systems design and implementation, 1999, pp. 173–186.

[141] Jean-Philippe Martin, Lorenzo Alvisi, Fast Byzantine consensus (PDF), IEEE Trans. Dependable Secure Comput. 3 (3) (2006) 202–215.

[142] Eli Gafni, Leslie Lamport, Disk paxos, 2002, available at: https://lamport.azurewebsites.net/pubs/disk-paxos.pdf.

[143] Leslie Lamport, Dahlia Malkhi, Lidong Zhou, Vertical paxos and primary-backup replication, in: PODC '09: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing, 2009, pp. 312–313.

[144] Iulian Moraru, David G. Andersen, Michael Kaminsky, Egalitarian Paxos, available at: https://www.pdl.cmu.edu/PDL-FTP/associated/CMU-PDL-12-108.pdf.

[145] Yanhua Mao, Flavio P. Junqueira, Keith Marzullo, Mencius: building efficient replicated state machines for WANs, in: Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation (OSDI'08), USENIX Association, USA, 2008, pp. 369–384.

[146] Dahlia Malkhi, Leslie Lamport, Lidong Zhou, Stoppable Paxos, available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2008/04/stoppableV9.pdf.

[147] Cheng Wang, Jianyu Jiang, Xusheng Chen, Ning Yi, Heming Cui, ApUS: fast and scalable paxos on RDMA, in: Proceedings of the 2017 Symposium on Cloud Computing (SoCC '17), Association for Computing Machinery, New York, NY, USA, 2017, pp. 94–107.

[148] Intro to stellar - learn about stellar, 2020, [Online]. Available: Accessed:09-Feb-2020.

[149] N. Payments, Mobilecoin, 2017, pp. 1–4.

[150] Valentin Poirot, Beshr Al Nahas, Olaf Landsiedel, Paxos made wireless: Consensus in the air, in: Proceedings of International Conference on Embedded Wireless Systems and Networks (EWSN '19), USA, 2019, pp. 1–12.

[151] Raft consensus algorithm, 2020, [Online]. Available: Accessed:23-Jul-2020.

[152] Understanding the raft consensus algorithm: an academic article summary, 2020, [Online]. Available: Accessed:03-Sep-2020.

[153] Raft consensus algorithm - GeeksforGeeks, 2020, [Online]. Available: Accessed:04-Sep-2020.

[154] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm (extended version), 2014.

[155] Apache kudu in 5 minutes - lewis gavin - medium, 2020, [Online]. Available: Accessed:01-Sep-2020.

[156] Apache kudu - overview, 2020, [Online]. Available: Accessed:01-Sep-2020.

[157] W. Fu, X. Wei, S. Tong, An improved blockchain consensus algorithm based on raft, Arab J Sci Eng (2021).

[158] L. Hou, X. Xu, K. Zheng, X. Wang, An intelligent transaction migration scheme for RAFT-based private blockchain in internet of things applications, IEEE Commun. Lett. 25 (8) (2021) 2753–2757.

[159] A. Konnov, A. Makarov, M. Pozdnyakova, R. Safin, A. Salagaev, Practical Byzantine fault tolerance, in: Juv. Delinq. Eur. beyond Results Second Int. Self-Report Delinq. Study, No. February, 2010, pp. 359–368.
[160] B. Chase, E. MacBrough, Analysis of the XRP ledger consensus protocol, 2018.
[161] The dBFT algorithm, 2020, [Online]. Available: Accessed:09-Feb-2020.
[162] dBFT 2.0 algorithm, 2020, [Online]. Available: Accessed:09-Feb-2020.
[163] K. Lei, Q. Zhang, L. Xu, Z. Qi, Reputation-based Byzantine fault-tolerance for consortium blockchain, in: 2018 IEEE 24th Int. Conf. Parallel Distrib. Syst, 2018, pp. 604–611.
[164] C. Chen, J. Su, T. Kuo, K. Chen, Msig-BFT: A witness-based consensus algorithm for private blockchains, in: 2018 IEEE 24th Int. Conf. Parallel Distrib. Syst., 2018, pp. 992–997.
[165] Serguei Popov, William J. Buchanan, Fpc-BI: Fast probabilistic consensus within Byzantine infrastructures, 2019, available at: https://arxiv.org/pdf/1905.10895.pdf.
[166] Henrique Moniz, The Istanbul BFT consensus algorithm, 2020, eprint=2002.03613, 2020, available at: https://arxiv.org/abs/2002.03613.
[167] Hossam Samy, Ashraf Tammam, Ahmed Fahmy, Bahaa Hasan, Enhancing the performance of the blockchain consensus algorithm using multithreading technology, Ain Shams Eng. J. (2021) In press.
[168] M. Du, Q. Chen, X. Ma, Mbft: A new consensus algorithm for consortium blockchain, IEEE Access 8 (2020) 87665–87675.
[169] L6: Byzantine fault tolerance - YouTube, 2016, [Online]. Available: Accessed: 27-Jul-2020.
[170] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.
[171] Introduction to sawtooth PBFT – hyperledger, 2020, [Online]. Available: Accessed:15-Oct-2020.
[172] What is tendermint? | tendermint documentation, 2020, [Online]. Available: https://tendermint.com/docs/introduction/what-is-tendermint.html#tendermint-vs-x. [Accessed: 15-Sep-2020].
[173] Tendermint explained — Bringing BFT-based PoS to the public blockchain domain, 2020, [Online]. Available: Accessed:23-July-2020.
[174] Hyperledger fabric explainer video - YouTube, 2020, [Online]. Available: Accessed:31-May-2020.
[175] The difference between ripple and XRP | ripple, 2020, [Online]. Available: Accessed:26-Sep-2020.
[176] Xrp | ripple, 2020, [Online]. Available: Accessed:27-Sep-2020.
[177] About - neo smart economy, 2020, [Online]. Available: Accessed:09-Feb-2020.
[178] Coordicide announcing the simulation study of FPC, available at: https://blog.iota.org/simulation-study-of-fpc-5b9e8b5a6910/.
[179] Qin Wang, Jiangshan Yu, Shiping Chen, Yang Xiang, SoK: Diving into DAG-based Blockchain Systems, available at: https://arxiv.org/pdf/2012.06128.pdf.
[180] Yonatan Sompolinsky, Aviv Zohar, Accelerating bitcoin's transaction processing. fast money grows on trees, not chains, in: IACR Cryptology EPrint Archive, 2013(881), 2013.
[181] Yonatan Sompolinsky, Aviv Zohar, Secure high-rate transaction processing in bitcoin, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 507–527.
[182] Yoad Lewenberg, Yonatan Sompolinsky, Aviv Zohar, Inclusive block chain protocols, in: International Conference on Financial Cryptography and Data Security, Springer, 2015, pp. 528–547.
[183] S. Park, S. Oh, H. Kim, Performance analysis of DAG-based cryptocurrency, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), 2019, pp. 1–6.
[184] Y. Ding, H. Sato, Dagbase: A decentralized database platform using DAG-based consensus, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), 2020, pp. 798–807.
[185] L. Baird, He Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance, Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep, 2016.
[186] Fu Xiang, Wang Huaimin, Shi Peichang, Ouyang Xue, Zhang Xunhui, Jointgraph: A DAG-based efficient consensus algorithm for consortium blockchains, J. Software: Practice and Experience (2019).
[187] K. Gai, Z. Hu, L. Zhu, R. Wang, Z. Zhang, Blockchain meets DAG: A blockdag consensus mechanism, in: M. Qiu (Ed.), Algorithms and Architectures for Parallel Processing. ICA3PP 2020, in: Lecture Notes in Computer Science, vol. 12454, Springer, Cham, 2020.
[188] S.R. B., S. G. V. V., Ul-blockdag : Unsupervised learning based consensus protocol for blockchain, in: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), 2020, pp. 1243–1248.
[189] Tai-Yuan Chen, Wei-Ning Huang, Po-Chun Kuo, Hao Chung, Tzu-Wei Chao, DEXON : A Highly Scalable, Decentralized DAG-Based Consensus Algorithm, available at: https://eprint.iacr.org/2018/1112.pdf.
[190] Yonatan Sompolinsky, Yoad Lewenberg, Aviv Zohar, SPECTRE: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections, available at: https://eprint.iacr.org/2016/1159.pdf.
[191] Yonatan Sompolinsky, Shai Wyborski, Aviv Zohar, Phantom and GHOSTDAG a scalable generalization of nakamoto consensus, 2020, available at: https://eprint.iacr.org/2018/104.pdf.
[192] V. Buterin, A next-generation smart contract and decentralized application platform, Etherum (January) (2014) 1–36.
[193] S. Park, A. Kwon, G. Fuchsbauer, P. Gaˇ zi, J. Alwen, K. Pietrzak, SpaceMint: A Cryptocurrency Based on Proofs of Space, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), LNCS, vol. 10957, 2018, pp. 480–499.
[194] What is dash cryptocurrency?, 2020, [Online]. Available: Accessed:13-Feb-2020.
[195] N. Boumal, An introduction to litecoin, Quality (September) (2019) 1–9.
[196] Understanding EOS and delegated proof of stake — Steemit, 2020, [Online]. Available: Accessed:15-Feb-2020.
[197] Frequently asked questions | cockroachdb docs, 2020, [Online]. Available: Accessed:01-Jun-2020.
[198] Counterparty, 2020, [Online]. Available: Accessed:03-Sep-2020.
[199] Algorand protocol overview | algorand, 2020, [Online]. Available: Accessed:15-Feb-2020.
[200] About - GoChain gochain, 2020, [Online]. Available: Accessed:13-Feb-2020.
[201] M. Du, X. Ma, Z. Zhang, X. Wang, Q. Chen, A review on consensus algorithm of blockchain, in: 2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017, 2017-Janua, 2017, pp. 2567–2572.
[202] H.F. Ouattara, D. Ahmat, Blockchain consensus protocols towards a review of practical constraints, 2018, pp. 304–314, (2017).
[203] G.T. Nguyen, K. Kim, A survey about consensus algorithms used in blockchain, J. Inf. Process. Syst. 14 (1) (2018) 101–128.
[204] W. Wang, et al., A survey on consensus mechanisms and mining strategy management in blockchain networks, IEEE Access 7 (May) (2019) 22328–22370.
[205] M. Salimitari, M. Chatterjee, A survey on consensus protocols in blockchain for IoT networks, 2018, pp. 1–15.
[206] Z. Yu, X.G. Liu, G. Wang, A survey of consensus and incentive mechanism in blockchain derived from P2P, in: Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS, 2018-Decem, 2019, pp. 1010–1015.
[207] University of Engineering and Technology, ICOSST: 2018 international conference on open source systems and technologies: proceedings: 19-21 december, 2018, lahore, Pakistan, in: 2018 12th Int. Conf. Open Source Syst. Technol., Al-Khawarizmi Institute of Computer Science, IEEE Computer Society. Lahore Section, IEEE Lahore Section, and Institute of Electrical and Electronics Engineers, 2018, pp. 54–63.
[208] K. Sharma, D. Jain, Consensus algorithms in blockchain technology: A survey, in: 2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019, 2019, pp. 1–7.
[209] S.J. Alsunaidi, F.A. Alhaidari, A survey of consensus algorithms for blockchain technology, in: 2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019, 2019, pp. 1–6.
[210] S.S. Panda, B.K. Mohanta, U. Satapathy, D. Jena, D. Gountia, T.K. Patra, Study of blockchain based decentralized consensus algorithms, in: IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, 2019-Octob, 2019, pp. 908–913.
[211] H.S. Jennath, S. Asharaf, Survey on Blockchain Consensus Strategies, in: Lecture Notes in Electrical Engineering, vol. 601, 2020, pp. 637–654.
[212] A. Andrey, C. Petr, Review of existing consensus algorithms blockchain, in: Proc. 2019 IEEE Int. Conf. Qual. Manag. Transp. Inf. Secur. Inf. Technol. IT QM IS 2019, 2019, pp. 124–127.
[213] Y. Hu, K.K.M. Editors, Advances in intelligent systems and computing 1097 ambient communications and computer systems, 2019.
[214] S. Hattab, I.F. Taha Alyaseen, Consensus algorithms blockchain: A comparative study, Int. J. Perceptive Cogn. Comput. 5 (2) (2019) 66–71.
[215] Y. Xiao, N. Zhang, W. Lou, Y.T. Hou, A survey of distributed consensus protocols for blockchain networks, IEEE Commun. Surv. Tutorials 22 (2) (2020) 1432–1465.
[216] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P.K. Singh, W.C. Hong, A survey on decentralized consensus mechanisms for cyber physical systems, IEEE Access 8 (2020) 54371–54401.
[217] P.B. Honnavalli, A.S. Cholin, A. Pai, A study on recent trends of consensus algorithms for private blockchain network, 2020, pp. 31–41.
[218] G.R. Carrara, L.M. Burle, D.S.V. Medeiros, C.V.N. de Albuquerque, D.M.F. Mattos, Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking, Ann. Des Telecommun. Telecommun. 75 (3–4) (2020) 163–174.
[219] M.S. Ferdous, M.J.M. Chowdhury, M.A. Hoque, A. Colman, Blockchain consensus algorithms: A survey, 2020, pp. 1–39.
[220] M. Salimitari, M. Chatterjee, Y.P. Fallah, A survey on consensus methods in blockchain for resource-constrained IoT networks, Internet of Things 11 (2020) 100212.
[221] J. Khamar, H. Patel, An Extensive Survey on Consensus Mechanisms for Blockchain Technology, in: Lecture Notes on Data Engineering and Communications Technologies, vol. 52, Springer, 2020, pp. 363–374.
[222] Sarah Bouraga, A taxonomy of blockchain consensus protocols: A survey and classification framework, Expert Syst. Appl. 168 (2021).