IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 27, 2019, accepted January 12, 2020, date of publication January 17, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967218

Solutions to Scalability of Blockchain: A Survey

QIHENG ZHOU, HUAWEI HUANG, (Member, IEEE), ZIBIN ZHENG, (Senior Member, IEEE), AND JING BIAN

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510275, China

Corresponding authors: Huawei Huang (huanghw28@mail.sysu.edu.cn) and Zibin Zheng (zhzibin@mail.sysu.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2016YFB1000101, in part by the National Natural Science Foundation of China under Grant 61902445 and Grant 61722214, and in part by the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2016).

ABSTRACT Blockchain-based decentralized cryptocurrencies have drawn much attention and been widely-deployed in recent years. Bitcoin, the first application of blockchain, achieves great success and promotes more development in this field. However, Bitcoin encounters performance problems of low throughput and high transaction latency. Other cryptocurrencies based on proof-of-work also inherit the flaws, leading to more concerns about the scalability of blockchain. This paper attempts to cover the existing scaling solutions for blockchain and classify them by level. In addition, we make comparisons between different methods and list some potential directions for solving the scalability problem of blockchain.

INDEX TERMS Blockchain, scalability.

I. INTRODUCTION

Blockchain as an emerging technology to realizing the distributed ledgers has attracted extensive research attention recently. Such a ledger intends to achieve decentralized transaction management, which means that any node joining the ledger can initiate transactions equally according to rules, and the transaction does not need to be managed by any third party. All transactions in the system are stored in blocks, which are then linked as a chain and organized in chronological order. Moreover, transactions that have written in blocks are immutable and transparent to all peers. With all these attractive characteristics, blockchain is drastically different from the traditional centralized trust entities and becomes a significant enabler to future financial systems. In recent years, the blockchain has developed rapidly, from Bitcoin [1], the first decentralized cryptocurrency, to Ethereum [2] with smart contracts, followed by the emerging permissioned blockchain (e.g. Hyperledger fabric [3]). Because of the wide adoption of Blockchain, blockchain based applications have been getting involved in our daily lives.

When the number of users of blockchain systems increases extensively, the scalability issues of major public-chain [4] platforms (e.g. Bitcoin and Ethereum) have arisen and greatly affected the development of blockchain.

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

Transaction throughput and transaction confirmation latency are two most talked-about performance metrics of blockchain and both of them have not reached a satisfactory level in recent popular blockchain systems [5], which leads to the bad user's quality of experience. However, compared with the centralized payment system like banks, these two metrics cannot be improved easily in blockchain, a self-regulating system, that needs more considerations in order to maintain decentralization. After numerous studies on the particularities of blockchain, some researchers raise the view of Blockchain Trilemma [6]. Similar to the CAP theory [7] in the traditional field of the distributed system, the Blockchain Trilemma points out that three important properties of a blockchain system, involving decentralization, security, and scalability, cannot perfectly co-exist. For instance, considering a simplified circumstance, adding a centralized coordinator into the system can reduce the consumption (e.g. computational resources consumed by proof-of-work [8]) for all users in the system to reach consensus on a set of transactions. Another example, shortening the block interval of Bitcoin can increase the transaction throughput but also affects the security of the whole system because of the increasing probability of fork. Therefore, balancing or even achieve these three aspects of blockchain system well is essential for the future development of blockchain that is suitable for more complex and larger-scale scenes in our daily lives.

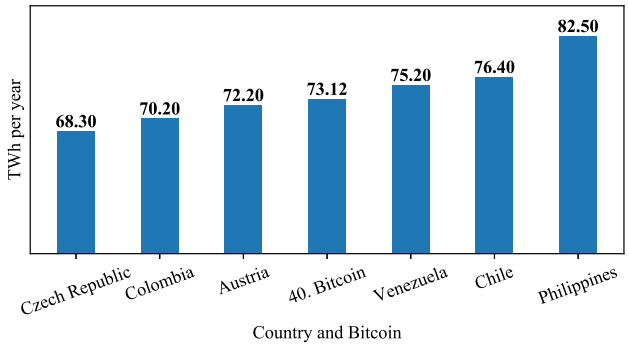

FIGURE 1. Energy Consumption by Country and Bitcoin: the number above the bar indicates the energy consumption and Bitcoin consumes 73.12 TWh per year, which ranks the 40th among all countries.) Source: https://digiconomist.net/bitcoin-energy-consumption.

In order to improve the scalability of the blockchain, many companies and research teams have proposed a large number of different solutions. We classify them according to the hierarchical structure of blockchain. In detail, the hierarchical structure mainly includes two layers, which are described briefly as follows. Layer1 concentrates on the on-chain design of blockchain including the structure of blocks, consensus algorithm and also the specific structure of the main-chain. On the other hand, Layer2 focuses on off-chain methods, which intends to reduce the burden of the main-chain, such as executing some transactions off-chain and moving some complex computational tasks to an off-chain platform. Layer1 (on-chain) solutions such as Bitcoin-Cash [9] increasing the block size, *Compact block relay [10]* compressing the blocks, *Sharding* techniques [11]–[14], and various improved consensus algorithms [15]–[19], in which the transaction throughput is increased and transaction latency is decreased, respectively. Layer2 solutions like *payment channel* (Bitcoin's Lightning network [20] and *side chain* (Plasma [21] of Ethereum) are still under developing. The *cross-chain* solutions that emerged in the last few years also play an important role in Layer2 scaling solutions. One of the most representative solutions is *Cosmos* [22], which aims to connect multiple independent blockchains to establish an integrated blockchain network and achieve scalability. Although the existing solutions somewhat improve the scalability, it should be noticed that most of these solutions sacrifice the most fundamental property of blockchain, i.e., decentralization, and also bring new security issues. In summary, with both advantages and limitations, those solutions are striving to achieve decentralization, security, and scalability simultaneously.

The rapid development of blockchain technology has drawn growing attention. However, its performance still needs much improvement compared with the mainstream payment processors such as Visa. Therefore, to accelerate the wide adoption of blockchain technology, the scalability issue requires many other more sophisticated solutions in the future.

In this paper, we intend to classify various existing scalability solutions towards blockchain.

The rest of this paper is organized as follows. Section II provides some concrete facts to briefly explain the performance of several typical blockchains. Section III presents the solutions to the scalability issue of blockchain proposed in recent years. Then, section IV discusses some open issues and future directions to scale blockchain. Finally, section V summarizes this paper.

II. SCALABILITY ISSUE OF BLOCKCHAIN

With the domination of Bitcoin in cryptocurrency, the scalability issues of blockchain have been exposed, too. Kyle Croman et al. [23] analyzed several key metrics to measure the scalability of Bitcoin, including *maximum throughput*, *latency*, *bootstrap time* and *cost per confirmed transaction (CPCT)*. The maximum throughput and latency are the two most important performance metrics that have a significant impact on the user's quality of experience (QoE).

Among all metrics listed above, transaction throughput receives the most attention. It has been reported that Bitcoin's highest transaction throughput is 7 TPS (transaction-per-second) [24] while Visa can achieve more than 4000 TPS [25] Obviously, low throughput of Bitcoin cannot satisfy the large-scale trading scenarios.

In theory, transaction throughput is restrained by the block interval and the block size. A larger block can store more transactions, directly raising throughput, but it also causes an increase in block propagation time. To ensure the current block to be propagated to most peers in the whole network before the next block is generated, which is critical to reducing the probability of *fork*, the block size and the average block interval between two successive blocks should be well configured. In Bitcoin, the block interval is about 10 minutes, and the block size is around 1 MB [1], which limits the number of transactions that can be stored in each block. Thus, to maintain the block propagation time while increasing the block size, the average bandwidth of the whole system that determines the block propagation time becomes a performance bottleneck of the blockchain system.

Another metric, transaction confirmation latency that is the time for a transaction to be confirmed, also has a strong relation with user experience.

Due to the huge volume of Bitcoin transactions nowadays, the limited size of blocks is far from enough to deliver all transactions submitted by nodes. Under such a situation, miners tend to select transactions that are with high transaction fees. As a result, the transactions that are with a low bid have to wait until packaged, which leads to the longer transaction latency [50]. Ethereum, another popular PoW-featured blockchain (in its pre-2.0 version) makes this problem even severe since some popular decentralized applications (DApps) [51] in Ethereum have induced extensive congestion in the entire network. As we can see in Figure 2, the total number of Ethereum transactions waiting to be confirmed in a certain period maintains a high level.

TABLE 1. Taxonomy of the scalability solutions in different layers.

Layer	Categories	Solutions
Layer2: *Non On-Chain*	Payment channel	Lightning Network [20], DMC [26] Raiden Network [27], Sprites [28]
	Side chain	Pegged Sidechain [29], Plasma [21] liquidity.network [30]
	Cross-chain	Cosmos [22], Polkadot [31]
	Off-chain computation	Truebit [32], Arbitrum [33]
Layer1: *On-Chain*	Block data	SegWit [34], Bitcoin-Cash [9] Compact block relay [10], Txilm [35] CUB [36], Jidar [37]
	Consensus	Bitcoin-NG [15], Algorand [16] Snow white [17],Ouroboros [18] [19]
	Sharding	Elastico [11], OmniLedger [12] RapidChain [13],Monoxide [14]
	DAG	Inclusive [38], SPECTRE [39] PHANTOM [40], Conflux [41] Dagcoin [42], IOTA [43] Byteball [44], Nano [45]
Layer0	Data propagation	Erlay [46], Kadcast [47] Velocity [48],bloXroute [49]

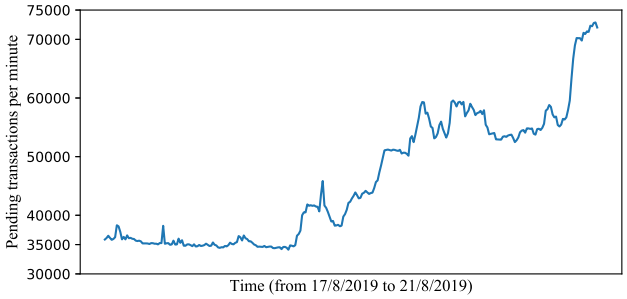

FIGURE 2. Ethereum pending transactions queue: The number of Ethereum pending transactions in a certain period. Source: Etherscan.io.

Besides the performance bottleneck of blockchain, we should also consider the capacity problem of blockchain seriously. As the scale of a blockchain growing rapidly, the storage required by all blocks grows accordingly. Therefore, the full nodes, which store all block data of the network, are required large storage capacity for each. Similarly, the Bootstrap time will increases linearly as the blockchain history grows, slowing down the process of new nodes joining into the system. All these restrictions degrade the availability and decentralization of a blockchain, and thus should be examined closely when developing a large-scale blockchain. Nowadays, more block compression methods have been proposed to reduce redundant data of blocks, which is beneficial for easing the capacity problem. At the same time, sharding techniques, partitioning the whole blockchain network into different shards, have been researched more detailed to solve the capacity problem of blockchain.

Meanwhile, many concerns have been raised about the energy consumption of Proof-of-work based blockchain systems, such as Bitcoin and Ethereum [52]. Miners in a PoW-featured blockchain are always competing with each other through calculating, which results in a large dissipation of electricity. Figure 1 shows the energy consumption of Bitcoin comparing with that of some countries/states, where we can find that the entire Bitcoin network consumes even more energy than many countries, such as Austria and Colombia, and barely ranks the 40^{th}. Although PoW works securely, it's far from green enough to be a sustainable consensus mechanism for future blockchain.

Striving to improve the scalability of blockchains while maintaining security and decentralization, many existing approaches have been proposed by literature. We will review some mainstream solutions in the next section.

III. TAXONOMY OF THE APPROACHES TO SOLVING THE SCALABILITY OF BLOCKCHAIN

By Table 1, we classify the existing popular solutions of solving the scalability of blockchains into three layers: *Layer1* *Layer2*, and *Layer0*.

Layer1 focuses on consensus, network and data structure of blockchain, all of which are executed on-chain. In contrast, Layer2 seeks the opportunity to scale out blockchain by off-chain methods such as off-chain channel [20], [27], side-chain [21], [30] and cross-chain protocols [22], [31]. Besides, we also present a table 2 which shows the data of Transaction Per Second (TPS) and confirmation time of some representative scaling solutions.

In the subsequent parts of this section, we elaborate on these existing state-of-the-art solutions dedicated to improving the scalability of blockchains.

A. LAYER1: ON-CHAIN SOLUTIONS

1) SOLUTIONS RELATED TO BLOCK DATA

As discussed in Section2, the scalability problem has a certain relevance with block size. Obviously, increasing block size enables a block to include more transactions. Block

TABLE 2. Comparsion of transaction per second (TPS) and confirmation time among different solutions.

Project	Technology	TPS (tx/sec)	Confirmation time
Ouroboros [18]	PoS	257.6	2 min
ByzCoin [25]	PBFT	1000	15-20 sec
Algorand [16]	Byzantine agreement	875+	22 sec
RapidChain [13]	Sharding	7,380	8.7 sec
Monoxide [14]	Sharding	11,694	13-21 sec
Conflux [41]	DAG	6,400	4.5-7.4 min

compression can achieve the same effect and also reduce storage overhead. And, some other solutions explore methods to achieve data reduction are also proposed. In this section, we will introduce some approaches focused on these ideas.

a: SEGREGATED WITNESS

The Segregated Witness (SegWit) [34] defined in BIP141 [53] is designed to prevent non-intentional Bitcoin transaction malleability and to alleviate blockchain size-limitation that reduces the transaction speed of Bitcoin. It achieves the goals by splitting the transaction into two segments, removing the unlocking signatures from the original transaction hashes, and the new *Witness* structure will contain both the scripts and signatures.

SegWit also defines a new way to calculate the maximum block-size by assigning a weight for each block. The new calculation is shown as follows.

$$BW = 3 \times BS + TS,$$

where BW is new defined Block weight and BS is Base size including the size of the original transaction serialization without any witness-related data. TS stands for Total size, which is the size of transaction serialization described in BIP144 [53] Block weight is limited under 4 MB, and theoretically allowing more transactions can be accommodated in one block, which slightly increases the scalability performance of blockchain.

An additional design of SegWit is to provide convenience for deploying Lightning Networks [20], which will be introduced in the Part(B) of this section.

b: BITCOIN-CASH

In 2017, because of the scalability problem, Bitcoin experienced a hard fork [54] and was split into two blockchain branches, i.e., Bitcoin and Bitcoin-Cash. Bitcoin-Cash has increased its block size to 8MB, which is much larger than the size of its previous version (only 1MB in size). After that, Bitcoin-Cash was upgraded further, to expand the block size up to 32MB. The average block interval of Bitcoin-Cash is still maintained at the original 10 minutes. In theory, the transaction throughput can be greatly increased. This has been verified in the stress test conducted in September 2018.

From the theoretical and practical points of view, improving the block size can scale-out the blockchain capacity directly. However, the infinite expansion enlarges the size of each block, which cannot be transferred easily due to the limitation of intra-blockchain bandwidth. Thus, only increasing the block size is not a sustainable solution. Some other studies [55], [56] also claim that larger blocks may lead to the problem of centralization since individual users in the network are not able to propagate blocks efficiently and also have difficulty in verifying a large number of transactions within a given interval. This will result in that only a centralized organization can act as a full node.

c: BLOCK COMPRESSION

To improve the throughput of blockchains, various solutions related to block compression have been proposed (e.g. Compact block relay [10]and Txilm [35]). All these methods share a similar idea that is to reduce some redundant data of a block that has been already stored in the *Mempool* of receivers.

Compact block relay was proposed in BIP152 [10] and altered the data structure of origin blocks in Bitcoin. A compact block contains the header of the block and some short transaction IDs (TXIDs) which will be used for matching transactions that are already available to the receivers.

Figure 3 shows the workflow of this protocol. BIP152 provides two modes for block relay. The essential part of the protocol is sending *cmpctblock* messages and receivers dealing with the messages. Node A send a compact block to Node B. The moment Node B receives the block, Node B should calculate TXIDs of the transactions in their Mempool and match each of them with TXIDs stored in the compact block. Then, if all unconfirmed transactions are available to Node B, the full block can be reconstructed. Otherwise, Node B should send a *getblocktxn* message to require the information of transactions they do not have and reconstruct the block after they receive all the data they need. The main difference between the provided two modes is that, in Low Bandwidth Relaying, the compact blocks are sent only if the receivers make requests.

Txilm is a protocol based on BIP152 [10] that compresses transactions in each block to save the bandwidth of the network. Txilm utilizes a short hash of TXID to represent a transaction, which achieves a greater result on block compression. However, hash collisions are more likely to occur when a short hash is used. Therefore, Txilm optimizes the protocol using sorted transactions based on TXIDs to

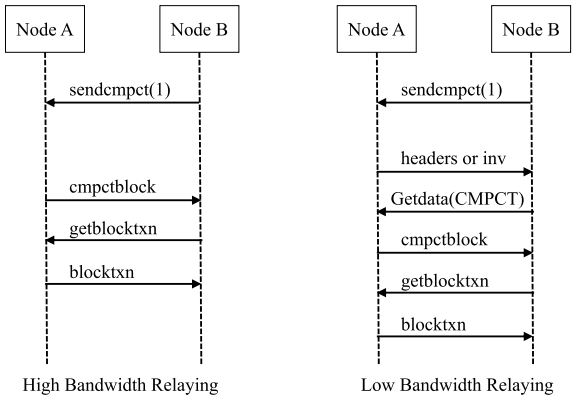

FIGURE 3. Procedures of two modes of compact block relay.

reduce the probability of hash collision and prevent the system from Collision attack by adding "SALT" (e.g. CRC32-Merkle root) when computing the hash of TXIDs. Based on the protocol, 80 times of data reduction is realized in their simulations and thus increases the throughput of the blockchain.

Some other approaches [57], [58], concerned with data of the block is also proposed in recent years. All existing solutions make some contributions to increasing the transaction throughput of blockchain but demand more optimization to scale the blockchain system.

d: STORAGE SCHEME OPTIMIZATION

Apart from block compression, there are some other solutions to reduce the storage pressure of each user.

CUB [36] proposes a scheme that assigns different nodes into a *Consensus Unit*. In each unit, each node stores part of the block data. The blocks of the whole chain are assigned to nodes in the unit to minimize the total query cost. They name this process as *block assignment problem* and propose algorithms to solve it, which reduces the storage overhead of each node while ensuring the throughput and latency.

Jidar [37] is a data reduction approach for Bitcoin system. The main idea of Jidar is to allow users only store relevant data they are interested in and thus releases the storage pressure of each node. When a new block is created, each node stores only a small part of the total data of a block, including relevant transactions and Merkle branch. Jidar adopts bloom filter to validate if the input of a transaction has been spent. Besides, if some users want to get all block data of the system, they can ask other nodes for data and cohere all fragments into complete blocks. However, incentive mechanisms are required to support this function.

2) DIFFERENT CONSENSUS STRATEGIES

We then review different consensus strategies of blockchain and some optimizations proposed to improve the scalability of blockchain.

a: PoW (PROOF OF WORK)

Bitcoin, proposed in 2008, adopted the PoW to achieve consensus in a decentralized network [1]. Under PoW, participants, also called *miners*, need to solve a computational task in order to generate a new block. When the answer is found, the miner broadcasts a relevant message to the network for other miners to verify the new block. If the block is validated, it can be added into the chain and the miner who generates it will be rewarded with tokens such as bitcoins. PoW is a novel consensus and has been exploited by a large number of blockchains. However, since Bitcoin has the risk to suffer from forks, transaction confirmation time is set to around one hour (after 6 blocks being mined). Even worse, the calculation in PoW has led to too much resource dissipation.

Therefore, some other studies [15], [39], [59] were dedicated to improving the original PoW mechanism. For example, Bitcoin-NG [15] is a blockchain protocol based on Nakamoto consensus [1]. It divides time into epochs, and each epoch has a single leader responsible for transaction serialization. In order to support this mechanism, Bitcoin-NG introduces two types of blocks: key block and microblock. The key block, generated by the miners through the PoW mechanism, does not contain transaction data and is only used for the election of the leader. And, the leader is allowed to generate the microblock which contains the packaged transaction data. Thus, transactions can be processed continually until the next leader is elected that significantly reduces transaction confirmation time and improves the scalability. GHOST [59] also builds upon PoW and re-organizes the data structure of Bitcoin to eliminate the security concern of double-spending [8] attacks, spending the same asset more than once, caused by network delay. SPECTRE [39] is a PoW-based protocol that utilizes the structure called direct acyclic graph (DAG) to improve the transaction throughput and reduce the confirmation time of Bitcoin.

b: PoS (PROOF OF STAKE)

PoS is an alternative mechanism that avoids the computational overhead of PoW. Instead of consuming computational resources to get involved in generating blocks, participants in PoS vote leaders by their investment in a blockchain system and thus reduce the confirmation time of transactions. The basic idea of PoS is that nodes with more currencies in the system are less likely to do harm to the system. However, because of the elimination of computational verification, to ensure the security of a PoS protocol is a challenging task. Many secure PoS protocols have been proposed. For example, Ouroboros [18] uses a coin-flipping protocol to elect leaders for the current epoch and seed for the next epoch. Participants in Ouroboros Praos [19] utilize a verifiable random function to generate a random number, which will be used to determine whether a participant can be elected as a leader. Snow-white [17] exploits a random oracle to elect a leader. Furthermore, Ethereum Casper [60] is planned to

release in 2020, which is equipped with a PoS protocol and is expected to improve the scalability of Ethereum.

c: DPoS (DELEGATED PROOF OF STAKE)

DPoS [61] is a new consensus protocol for blockchain and its principle is different from PoS. In DPoS, stakeholders elect a small group of delegates to be responsible for producing as well as validating blocks. DPoS has been adopted as the consensus algorithm for Bitshare [62] and EOS [63] to solve the problem of scalability. This algorithm is divided into two stages. The first stage is the staked voting, in which the nodes holding tokens can vote for the potential block producers, and finally, 21 producers with most votes are selected to create the next block. The idea is to let the token holders in the network vote for producers who can provide great computing power and indirectly vote the malicious nodes. A block is broadcast to other producers to be verified and if more than 15 block producers verify and sign, the block is confirmed. Such voting is continually performed throughout the system to select the producers, but if a selected block producer does not produce a block within 24 hours, it will be replaced by a spare producer. At the same time, the probability of this producer to be selected in the future will be reduced as well because of its previous failures.

In EOS, a block is generated by one producer every three seconds on average, and the average confirmation time for each transaction is about 1.5 seconds. Compared with other mainstream blockchain platforms, EOS can reach an overwhelming million-level TPS. However, its decentralization has been questioned. It is believed that more than 50% of the coins in EOS are occupied by only ten addresses, and less than 1% of EOS addresses hold more than 86% tokens of EOS [64]. The DPoS applied by EOS actually chooses the super node that holds the most resources, resulting in the rights are in the hands of a small number of nodes, which is essentially viewed as a centralized mode.

d: PBFT (PRACTICAL BYZANTINE FAULT TOLERANCE)

PBFT [65] is a replication algorithm that is able to tolerate the Byzantine faults [66], consistency problems caused by unreliable components or nodes in the system, in asynchronous systems and performs more efficiently than early approaches [67]–[69]. In every view of PBFT, a primary server is selected to be responsible to order messages. When primary receives a client request, a three-phase protocol begins working, including *pre-prepare*, *prepare*, *commit* phases. In the *pre-prepare* phase, primary broadcasts the *pre-prepare* messages in an ordered sequence to other replicas. In the *prepare* phase, each server makes a choice to accept the *pre-prepare* message or not. If accepted, the server broadcasts a *prepare* message to all other replicas. When it successfully collect $2f + 1$ feedback messages (f indicates the number of faulty nodes), it starts the *commit* phase. Similar to the *prepare* phase, each server broadcasts *commit* messages to others and waits for $2f + 1$ feedback messages from other replicas which indicates that a majority of servers agree to accept the client's request and send a reply to the client.

In contrast to PoW, PBFT works without computational tasks. It thus reduces the complexity of consensus to the polynomial level but requires more communication overhead. Some follow-up works build their consensus protocols based on PBFT and make some modifications. For example, Tendermint [70] uses validators with voting power to vote for each round and reach consensus finally. Elastico [11] is a sharding protocol that chooses PBFT as the consensus for each committee of Elastico to agree on a single set of transactions.)

e: HYBRID CONSENSUS

Hybrid Consensus is a protocol that combines some classical consensus protocols. ByzCoin [25] proposes a two-phase protocol based on the idea of Bitcoin-NG [15]. However, it is able to ensure strong consistency by combining PoW and PBFT. In addition, ByzCoin uses a collective signing protocol called Cosi [71] to reduce the cost of the *prepare* and *commit* phases of PBFT and scale it to large consensus groups. Later works such as Hybrid consensus [72], Solidus [73] also propose to combine different protocols with PoW aiming to improve on the throughput and security.

Algorand [16] is a cryptocurrency based on a Byzantine Agreement (BA) protocol. By combining with Verifiable Random Functions [74], users are chosen to become a committee member to participate in the BA and reach consensus on the next set of transactions. To mitigate targeted attacks, the participant will be replaced after sending a message in BA. With all these approaches, Algorand scales to 500,000 users in experiments and achieves high throughput.

f: OTHER CONSENSUS

Some other new consensus algorithms have been proposed in recent years, including PoA (proof-of-authority) [75], PoC (proof of capacity) [76] and PoP (proof-of-Participation) [77], which make some modifications of the previous consensus to improve the scalability of blockchain.

PoP (Proof of Participation) is a new protocol that implements PoS through the mining mechanism of PoW. PoP selects a list of stakeholders to work out a computational task, which is simpler than that in PoW, to generate a new block. Other stakeholders who did not participate in the mining validate the block and propagate it. Unlike PoS, transaction fees in PoP are only distributed to stakeholders participating in validation and propagation, which thus encourages stakeholders to maintain an online node and sustain the system. PoP includes two layers of security, proof-of-work, and proof-of-stake, that protect the system from security problems (e.g. double-spending) and also consume less energy than the traditional PoW mechanism.

PoC (Proof of Capacity) is a consensus algorithm that utilizes the storage resource (disk space) to mine. Miners in PoC based system stores a list of possible answers before mining. Larger space indicates a higher possibility of generating the

next block and getting the reward. PoC is similar to PoW but reduces energy consumption by complex computational tasks.

PoA (Proof of Authority) is a modified form of PoS where a block validator's identity plays the role of stake and relies on a set of selected validators to reach consensus. Since a new block is validated by authorized nodes, a small part of nodes in the network, the speed of validating processes is highly increased. PoA is suitable for permissioned blockchain where nodes' identities are authorized and increases the performance in terms of the TPS.

3) SHARDING

Sharding [78] is a traditional technology first proposed in the database field mainly for the optimization of large commercial databases. This method is to divide the data of a large database into a number of fragments, and then store them in separate servers to reduce the pressure of a centralized server, thereby improving the search performance and enlarging the storage capacity) of the entire database system.

The basic idea of sharding technology is *divide-and-conquer*. Therefore, applying sharding technology to blockchain is to divide a blockchain network into several smaller networks, each contains a part of nodes, which is called a ''shard''. Transactions in the network will be processed in different shards, so that each node only needs to process a small part of arriving transactions. Different shards can process transactions in parallel, which can boost the concurrency of transaction processing and verification, thus increasing the throughput of the entire network. While partitioning the whole system into different shards, it is critical to protect the decentralization and security of the system. Several aspects required to particularly take into account: (a) How to reach a consensus in each shard and prevent each shard from suffering some common risks such as 51% vulnerability and Double-spending. (b) How to handle cross-shard transactions quickly while ensuring the consistency of these transactions.

Figure 4 shows an example of sharding architecture, where the blockchain network is divided into 3 shards, including three procedures:

- At first, peers in the network are assigned to different shards. In order to reduce the storage overhead of each node, *State sharding* enables nodes in each shard only need to store the state of their own shard.
- *Transaction sharding* distributes transactions to different shards and allow transactions to be processed in parallel. Apart from transactions executed within a single shard, cross-shard transactions are very common in a large system. Therefore, the system should be equipped with some protocols to deal with cross-shard transactions carefully and efficiently.

As cross-shard transactions require more communication costs and also increase the confirmation latency, transactions in a sharding-based system should be placed into shards more carefully based on some partitioning rules. This process should consider different factors including the balance among different shards, the possible number of cross-shard transactions and the total amount of data that would be reallocated when rescheduling shards [79]. Some classical graph partitioning algorithms can be adopted, such as Kernighan-Lin algorithm [80] and METIS [81]. Hashing is another straightforward approach that uses the hash result of the unique id of each account as the id of the selected shard.

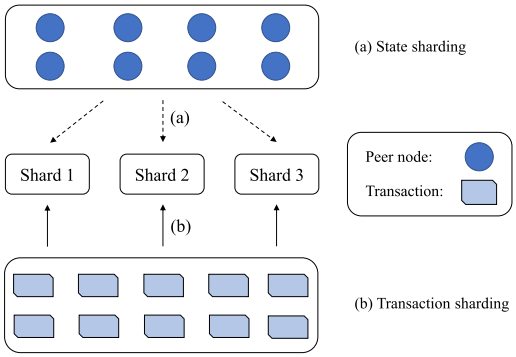

FIGURE 4. Illustration of sharding. The initial network contains eight nodes (blue circle). After (a), nodes are assigned to different shards. (b) Transactions are distributed to different shards and be processed in parallel.

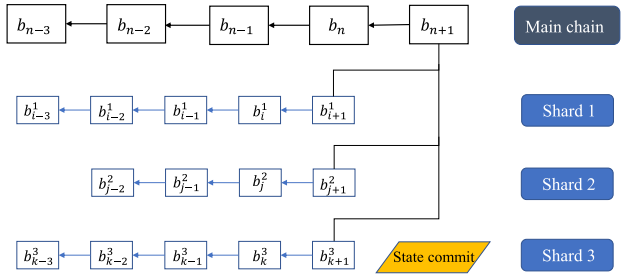

FIGURE 5. Architecture of the sharding protocol with a main chain.

Some other solutions proposed a new structure consisting of a main chain and multiple shard chains. Each shard maintains a shard chain and commits its state to the main chain periodically. From the architecture shown in Figure 5, we can see that each shard has a dedicated chain. Under this kind of architecture, cross-shard transactions are processed through the main chain by admitting the *receipts* of cross-shard transactions committed by different shards, which can be validated by all shards to ensure the correctness of cross-shard transactions. However, when the scale of cross-shard transactions increases in a blockchain system, the main chain will become the bottleneck of the holistic system since the large volume of transactions brings great pressure of both storage and communications.

We also find that several existing works [11]–[14] have exploited various methods to optimize their systems based on the sharding technology. Each of those representative works is reviewed as follows.

a: ELASTICO

Elastico [11] is the first sharding protocol for the permission-less blockchain. In each consensus epoch of Elatico, participants need to solve a PoW puzzle, which will be used to determine the consensus committee. Every committee works as a shard and runs PBFT [65] to reach the consensus and the result will be committed to a leader committee, which is responsible for generating the final decisions on the consensus results of other shards. Finally, the final value will be sent back to update other shards. However, there are several drawbacks of Elastico:

- Elastico generates identities and committees in each epoch. Such frequent operation potentially degrades the efficiency of transaction execution.
- Although each node only needs to verify transactions within its own shard, each node is still required to store all data of the entire network.
- Elastico requires a small size to limit the overhead of running PBFT in each committee, leading to a high failure probability while only tolerating up to a 1/4 fraction faulty nodes.
- Elastico fails to ensure the cross-shard transaction atomicity.

b: OMNILEDGER

OmniLedger [12], a more recent distributed ledger based on Sharding technique, builds closely on Elastico [11] and tries to solve the problems of Elastico. It uses a bias-resistant public-randomness protocol for shard assignment, which combines RandHound [82] with Algorand [16]. To guarantee the atomicity of cross-shard transactions, OmniLedger introduces a two-phase client-driven ''lock/unlock'' protocol called Atomix. OmniLedger also adopts the data structure *blockDAG* [38] to make block commitment parallelly and increase transaction throughput via *Trust-but-Verify Validation*. However, the following issues still remain unsolved in OmniLedger:

- Similar to Elastico, OmniLedger is also resilient to Byzantine faults only up to a 1/4 fraction.
- Users in OmniLedger are required to participate actively in cross-shard transactions, which is very difficult to satisfy light-weight users [83]

c: RapidChain

RapidChain [13] is a sharding-based public blockchain protocol that is resilient to Byzantine faults up to a 1/3 fraction of the participants, which is better than the 1/4 fraction of OmniLedger [12]. RapidChain reveals that the communication overhead per transaction is a major bottleneck to the transaction throughput and latency in previous sharding-based protocols [11], [12]. Therefore, Rapidchain reduces the amount of data exchange per transaction and does not need to gossip transactions to the entire network because of the usage of a fast cross-shard verification technique. Additionally, RapidChain utilizes block pipelining to reach a further improvement of throughput and ensures robustness via a reconfiguration mechanism.

d: MONOXIDE

Monoxide [14] is a scale-out blockchain that proposes *Asynchronous Consensus Zones* and scales the blockchain linearly while considerably maintaining decentralization and security of the system.

The entire network of Monoxide is divided into different parallel zones, each of which only needs to be responsible for itself since blocks and transactions are zone-specific and are only stored in their own zone. Handling transactions across shards (i.e., zones) is an essential issue in sharding-based blockchain systems. In Monoxide, *eventual atomicity* is proposed to ensure the correctness of cross-zone transactions. At the same time, Monoxide proposed an innovative *Chu-ko-nu Mining* that magnifies the mining power, enabling miners to create blocks in different zones via solving one PoW puzzle. Therefore, the difficulty of attacking a single zone is as difficult as attacking the entire network. This characteristic ensures the security of a single zone.

Some other public blockchain projects, including Zilliqa [84] and Harmony [85], also employed sharding technology to solve the scalability of their systems. Zilliqa is the first sharding-based public blockchain with PoW as the consensus algorithm. Zilliqa improves the TPS via processing transactions in different shards, but each node in Zilliqa still needs to store the data of the whole network, which will hinder the system to scale. Later, Harmony also adopts sharding to build a scalable and provably secure public blockchain. Harmony applies a structure with multiple Shard Chains, which processes transactions and store data within the shard, and a Beacon Chain that includes the block header from each Shard Chain and generates random numbers needed in the consensus. Besides, different from Zilliqa, Harmony divides the storage of blockchain data into different shards and a node in a shard only needs to store the data of its own shard.

At present, there are very few efficient sharding protocols that highly guarantee decentralization, scalability, and security. Thus, there remains a large research space for sharding technology.

4) DAG (DIRECTED ACYCLIC GRAPH)

The traditional blockchain stores transactions in blocks that are organized in a single chain structure. With this kind of structure, blocks cannot be generated concurrently and thus limits the transaction throughput. In order to solve this problem, an idea dedicated to revising the structure of blockchain called DAG [86] is proposed.

DAG is a finite directed graph with no directed cycles commonly used in the computer science field. An obvious way to transform blockchain into DAG is to let a block act as a vertex in DAG and connect to some previous vertices. However, different from blockchain, DAG allows several vertices to connect to a previous vertex which means concurrent block

TABLE 3. Comparsion between different DAG-based solutions.

Project	structure	consensus	block total order
Inclusive [38]	block DAG	PoW	No
SPECTRE [39]	block DAG	PoW	No
PHANTOM [40]	block DAG	PoW	Yes
Conflux [41]	block DAG (with pivot chain)	PoW (GHOST [11]) on a pivot chain	Yes
IOTA [43]	Tx DAG	Cumulative weight of transactions	No
Byteball [44]	Tx DAG	Relying on a reputable group called Witnesses	Yes
Nano [38]	Block-lattice	Balance-weighted votes on conflicting transactions	Yes

generation and thus enables more transactions to be included in the system.

Some representative proposals are briefly reviewed as follows. Y. Lewenberg et al. [38] utilize Directed Acyclic Graph of blocks (blockDAG) in their protocol. Different from the traditional structure of blockchain, in this protocol, a new block references multiple former blocks. An inclusive rule is proposed to select a main chain of the formed DAG. Moreover, the contents of off-chain blocks that do not conflict with previous blocks can also be included in the ledger. With the proposed protocol, the system achieves higher throughput.

Later, another DAG-based blockchain called SPECTRE [39] applies the DAG structure to represent an abstract vote to specify the partial order between each pair of blocks, which cannot be extended to a total order over all transactions.

PHANTOM [40] also applies blockDAG to achieve faster block generation and higher transaction throughput. Moreover, PHANTOM proposes a greedy algorithm to order transactions embedded in blockDAG and is able to support smart contract.

Conflux [41] is a fast and scalable blockchain system based on DAG. In Conflux, they proposed two different kinds of edges between blocks (i.e. parent edges and reference edges). A pivot chain formed by parent edges is selected via a selection algorithm. Therefore, the consensus problem of conflux is transformed to reach the consensus of a single chain, which they adopt GHOST [59] to solve.

In industry, there are also several DAG-based projects. A DAG-based cryptocurrency called Dagcoin [42] treats each transaction as a block and focuses on faster security confirmations and greater throughput. Similar to Dagcoin, another branch of studies aim to build DAG-based distributed ledgers, such as IOTA [43], Byteball [44] and Nano [45].

Fantom [87] proposed the *OPERA chain*, a DAG constructed by event blocks, and a Main-chain to determine the ordering between every block. *Lachesis Consensus* is also provided to reach faster consensus via more efficient broadcast.

In table 3, we make a comparison of selected properties (specific structure, consensus, whether ensuring total block order) among some DAG-based protocols. As the table shows, some of them aim at scaling the proof-of-work based system using DAG. And, the specific structure of them also has some differences between each other. Tx DAG stands

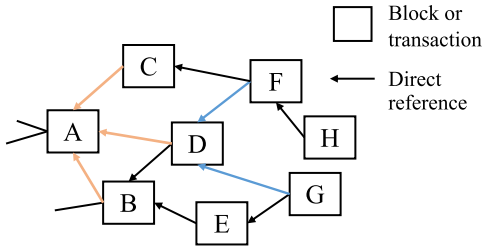

FIGURE 6. An overview of DAG: Each rectangle in the graph represents a block (or a transaction). Multiple blocks (or transactions) can be generated concurrently by linking to previous blocks (or transactions) in DAG (i.e. three orange arrows pointing to *A* and two blue arrows pointing to *D*).

for a DAG structure that is formed by many independent transactions that are not required to be packed into blocks. Total block order is an essential property that determines the order between every two blocks in the network and thus acts as an important role for protecting the system from several attacks (e.g. double-spending).

Tangle [88] is a DAG network under the basic idea of IOTA. As Figure 6 shows, Tangle is extended by adding directed edges between two transactions. Each edge represents that a new transaction has approved a previous transaction. In IOTA, there is no block, miner and transaction fee involved. Every node can create transactions freely after solving a specific computational task and choose two previous transactions to validate and approve them if valid. Later analysis [89], [90] also proves all these properties of Tangle. Besides, algorithms have been proposed to mitigate a kind of double-spending attack in Tangle called parasite chain attacks [91].

With such impressive merits, some other DAG cryptocurrency techniques have been proposed, like new randomized gossip protocol for consensus of Hashgraph [92] and the addition of DAG in Avalanche [93] to extend their consensus protocols, continuously improving the development of DAG.

Compared with blockchain, DAG-based platforms adopt a different ledger-structure and different transaction-confirming methods. However, some questions about IOTA are raised [94], focusing on the claimed great characteristics that IOTA do not need transaction fees and maintains high scalability. Meanwhile, treating each transaction as a block requires more metadata (e.g. reference to other vertices

in DAG) and thus cannot be applied as an efficient method for constructing a scalable system.

And, because of the consensus protocol utilized in some of the current DAG-based ledgers, security issues (e.g. double-spending [95]) and decentralization of these systems are controversial, which will probably limit the further development of DAG.

B. LAYER2: NON ON-CHAIN SOLUTIONS

We then classify the Layer-2 approaches into the following categories: *Payment Channel*, *Sidechain*, *off-chain computation*, and the *cross-chain*.

1) PAYMENT Channel

The payment channel is a temporary off-chain trading channel, transferring some transactions to this channel to achieve the effect of reducing the transaction volume of the main chain while improving the transaction throughput of the entire system. The representative payment channel solutions include Lightning network [20] adopted by Bitcoin, as well as the Ethereum-based Raiden network [27].

a: LIGHTNING NETWORK [20]

In recent years, the number of Bitcoin transactions has increased drastically, and its shortcomings have exposed, including high transaction delays and expensive transaction fees. To alleviate those drawbacks of the Bitcoin network, developers have proposed a new method - *lightning network*.

To explain briefly, the basic idea of Lightning Network is that two nodes in Bitcoin establish an off-chain trading channel, in which they can carry out multiple low-latency transactions. As shown in Figure 7, this solution includes three phases, establishing the channel, trading, and closing the channel. Before launching transactions, the two parties first have saved a certain amount of tokens in the channel as a deposit (greater than the total amount involved in the subsequent transaction), which is the first transaction to open the channel and is recorded on the Bitcoin main chain.

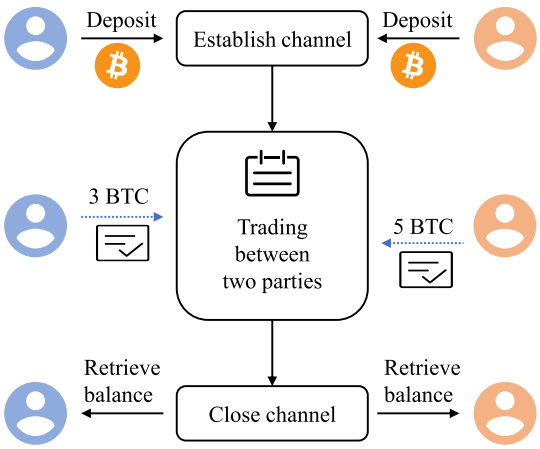

FIGURE 7. Procedures of lightning network.

Both parties can then trade with each other in the channel and if one of them cheats, all funds in this channel will be sent to a counterparty as penalty. When closing the channel, the amount of tokens on both sides is submitted to the block of the main chain. Therefore, multiple transactions are completed off-chain and the whole process produces only two transaction records submitted to the main chain. This approach greatly increases the number of transactions when the block size is a constant.

Furthermore, it is not necessary to establish a payment channel between every two parties who intend to exchange tokens. A Payment Channel Network (PCN) is introduced to conduct off-chain transactions between two parties that have no direct payment channel established between them. One participant to route to another via the path between them and make indirect transactions. Figure 8 shows the routing schematic diagram of the Lightning network. Node *0* and Node *9* establish a payment channel and carry out transactions directly. Node *1* is able to send transactions to Node *3* via the two channels (i.e. Node *1* to Node *2* and Node *2* to Node *3*). Similarly, Node *4* and Node *8* can trade with each other indirectly. Since transactions can only be sent through a route connected by different payment channels, a proper routing mechanism is needed to ensure the availability of Lightning Network, which has not been developed perfectly. Companies like Lightning Labs [96] implement protocols to build Lightning Network and help users make transactions freely.

Lightning Network provides instant and low-cost payment. However, the flaws of the lightning network are also very obvious. First, the off-chain channel requires both parties to be online at the same time. Second, it has been reported that the lightning network's large transaction success rate is low [97], indicating that current Lightning Network is not suitable for handling high-value transactions. These two disadvantages listed above greatly limit the wide-adoption of lightning networks.

b: RAIDEN NETWORK

Raiden Network is a payment-channel for Ethereum. Its implementation is very similar to the Lightning Network. The main difference is that the Raiden Network supports all ERC20 [98] tokens, while the Lightning Network is limited to Bitcoin transactions.

Payment channels have been widely researched in recent years, releasing several implementations of the Lightning Network [99]–[101]. Besides, there are many other solutions of off-chain payment channel from academia, including Bitcoin Duplex Micropayment Channels [26], Sprites [28], AMHLs [102]. Sprites develops constant locktimes to improve transaction throughput in Payment channel networks and support incremental deposits and withdrawals without interrupting the payment channel. AMHLS utilizes anonymous multi-hop locks to preserve privacy in the Payment channel and also reduce the communication overhead. There

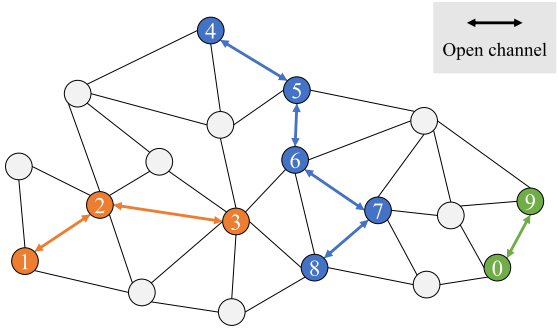

FIGURE 8. Lightning network topology: A circle represents a user in the lightning network and a left-right arrow indicates a trading channel established between both sides of the arrow.

remains a large space for research to provide a more effective and secure payment channel.

2) SIDECHAIN

Pegged Sidechain [29] is the first sidechain that enables assets in blockchains like Bitcoin to be transferred between different blockchains while preventing the assets from malicious attackers and also ensuring the atomicity of the transfers.

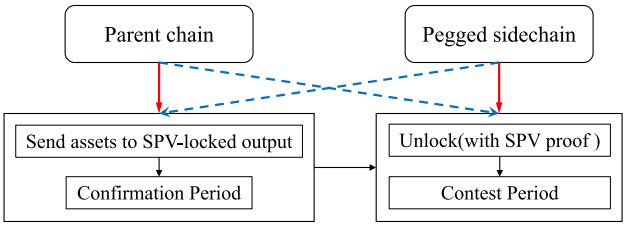

FIGURE 9. Two-way peg protocol of Pegged Sidechain [29]: Two red dotted lines indicate the procedure of transferring assets from the Parent chain to the Pegged sidechain. The blue dotted lines show the reverse procedure.

Figure 9 shows an example of transferring assets from parent chain to side chain by the *Two-way peg* protocol proposed in Pegged Sidechains [29]. First, the parent chain sends coins to a special output that cannot be unlocked without a Simplified Payment Verification (SPV) [103] proof on the pegged sidechain. After sending coins is a waiting period called *confirmation period*, which intends to protect the transferring from a denial of service attack and trades latency for security. Unlocking action is followed by the *contest period*, in which the newly-transferred assets cannot be spent on the sidechain, aims to prevent double-spending of the previously-locked assets.

Transferring assets from the Pegged sidechain back to the Parent chain is the same procedure as above, so the protocol is also called *Symmetric Two-Way Peg*.

a: PLASMA

Plasma [21] is a framework of sidechain attached to the Ethereum main chain. Its root is a smart contract running on the main chain, which records the rules and the state hash of the sidechain. Multiple child chains can be generated from the root, which is continuously expanding and finally become a tree structure. Users can create a ledger on the Plasma chain and achieve asset-transfer between the Plasma chain and the Ethereum main chain via the root. Users can also withdraw their funds from the chain any time.

Transactions can be carried out between different users on the child chains, similar to the situation under Bitcoin's Lightning Network. However, Plasma allows multiple participants to interact without requiring all participants to be online at the same time to update the transaction status.

Furthermore, Plasma can reduce the pressure of the Ethereum main chain by minimizing transaction status so that a simple hash can represent the update of multiple statuses. In this way, Plasma is capable to extend the transactions volume of the side chain.

While improving scalability, Plasma also provides some measures to ensure security avoid security hazards (e.g. double-spending) in sidechains. The Plasma chain submits the hash of the header of its block to the Ethereum main chain periodically. Thus, the main chain can verify the validity of transactions included in Plasma chains. If fraud is found in an invalid block, it will be rolled back with a slashed penalty.

Based on the framework aforementioned, many versions of Plasma have been designed. Minimal Viable Plasma (Plasma MVP) [104] is a simplified version based on the Unspent Transaction Outputs (UTXO) model and shows the fundamental properties of Plasma. Plasma Cash [105], a later improved version of Minimal Viable Plasma, proposes a mechanism in which each deposit operation corresponds to a unique coin ID and uses a data structure called Sparse Merkle Tree [106] to store the transaction history. Plasma Debit [107] is another implementation of Plasma framework and also an extension of Plasma Cash. Plasma is still under development and will be a potential solution to substantially scale out the blockchain systems.

b: LIQUIDITY NETWORK (NOCUST)

The previous *state-channel* solutions [20] require at least one transaction on the parent-chain when a channel is established, and also have the drawback that the transaction funds need to be saved in the trading channel as a deposit and the transaction channel relies on complex routing topologies.

The Liquidity.Network [30] team proposed the securely scalable commit-chain named Nocust [108], which has the following excellent properties:

- A new kind of data structure called Merkleized Interval Tree is a multi-layered tree. Individual user account balances are stored in exclusive non-crossing interval space, but the structure ensures that the balances of different users can be summed very quickly to verify whether the amount is the same as that recorded in the smart contract on the parent-chain.
- Nocust is non-custodial, that is, there is no need to limit the funds of the users on the chain, unlike the lightning

network which requires participants to deposit in prior for the channel.
- Users do not need to interact with the parent-chain to join the commit-chain. They are free to trade with each other, including transferring funds and receiving funds.
- Nocust can guarantee real-time transactions and reduce transaction delays without additional fees and mortgages.

The experimental results in the paper [108] show that Nocust can maintain a very low transaction fee and achieve a high transaction throughput when scaling to one billion users. These merits imply the practicality of its scalability solution.

3) OFF-CHAIN COMPUTATION

Miners in Ethereum need to emulate the execution of all contracts to verify their states. The process is costly and limits the scalability of Ethereum. Thus, some solutions have been proposed to build scalable smart contracts.

a: TRUEBIT

Truebit [32] is a system for verifiable computation that outsources complex computing tasks to an off-chain market. Such the off-chain market executes the tasks and verifies the results and finally submits them back to the main chain. It was originally designed to break the gas restrictions of the Smart Contracts in Ethereum platform. For instance, a DApp needs to perform a very complicated and expensive calculation task which is costly and inefficient in Ethereum. Then, the Truebit protocol is a good option for this DApp. Overall, Truebit is divided into three layers including the *Incentives Layer*, the *Dispute Resolution Layer*, and the *Computational Layer*. Each layer is elaborated as follows.

- *Computational Layer:* In this layer, users submit the computing task code and incentives to publish a task. There is an off-chain computing market, in which the miners listen to tasks and run the code after paying deposits. Each participant who solves a task is called a Solver, and each Verifier is responsible for verifying that a task is completed correctly.
- *Dispute Resolution Layer:* As the name suggests, Dispute Resolution layer is responsible for resolving disputes. When a computation is completed, the verifiers verify the result. If one of the verifiers finds that the result is incorrect, it can call into question about the result, and then both parties will be involved in a verification game. They can use interactive verification to find the specific steps they have in conflict.
  In the verification game, the party who is wrong will be punished, to prevent from deliberately cheating for both parties.
- *Incentives Layer:* Solvers get rewards by solving tasks and verifiers get rewards by detecting errors from the results computed by solvers. However, verifiers can't get a reward if no error found for a long time. If incentives for verifiers are not enough, the number of verifiers in the market will keep losing, resulting in the imbalance of the whole system. To solve this problem, Truebit adds a *forced error* mechanism that enforces the solvers to provide erroneous calculations periodically and add tags in the hash. In this way, when a verifier finds an error, both the solver and verifier can be rewarded, making verifiers profitable.

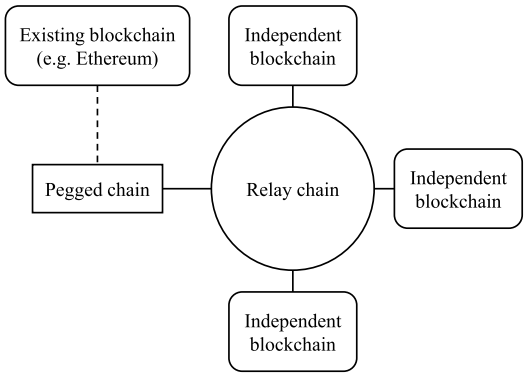

FIGURE 10. Architecture for relay [22], [31].

b: ARBITRUM

Arbitrum [33] introduces a new protocol that improves the scalability of smart contracts by moving the computation of verifying smart contracts off-chain. In Arbitrum, Verifier is a global role that validates transactions, e.g., Miners in Bitcoin. Arbitrum utilizes a Virtual Machine to implement a contract that owns a fund, which cannot be overspent by any execution of the contract. And, every party can create a VM and select a set of VM managers to force the VM to work correctly according to the VM's code. If all managers of a VM agree with the new state of VM, they sign a *Unanimous assertion*. On the other hand, VM managers sign a *Disputable assertion* to challenge the VM's state change and be engaged in the *bisection protocol*. The bisection protocol performs similarly with Dispute Resolution in Truebit, intending to determine if the VM's state change is correct. In this way, only hashes of contract states need to be verified by the Verifier. This releases the pressure of verifiers and also allows contracts to execute privately.

With the support of verifiable computation, large scale computation tasks can be solved off-chain, which provides great improvement in the scalability of blockchain systems.

4) CROSS-CHAIN TECHNIQUES

Nowadays, cross-chain projects are also fashionable and viewed as potential solutions to scale out blockchain systems.

Relay technique [22], [31] is another obvious idea of connecting different blockchains together, expecting to build a big network of blockchains and ensuring interoperability between different blockchains. Figure 10 shows a model of current inter chain architecture called Relay, the components of which includes *independent blockchains* built atop similar consensus and *relay chain* connecting all independent

blockchains. In addition, the *Pegged chain* (e.g. Peg Zone in Cosmos and Parachain bridge in Polkadot) is also provided to bridge existing blockchains with the cross-chain system.

Relay chain in Figure 10 serves the role as a router, enabling new independent blockchains to join in the cross-chain system and adopting cross-chain protocols to process cross-chain transactions more efficiently and also to ensure the consistency.

We then review several representative cross-chain projects as follows.

a: COSMOS

Cosmos [22] is an ecosystem of connected blockchains. The network is comprised of many independent blockchains, each of which is called a zone. Powered by consensus algorithms like Tendermint consensus, those zones can communicate with each other via their Inter-Blockchain Communication (IBC) protocol, allowing heterogeneous chains to exchange values (i.e. tokens) or data with each other. Hub (a framework like Relay-chain shown in Figure 10) is the first zone on Cosmos, and any other zones can connect to it. Therefore, Cosmos achieves inter-operability where zones can send to or receive from other zones securely and quickly via Hub, instead of creating connections between every two zones.

Cosmos also provides Tendermint core and Cosmos SDK (Software Development Kit) [109]) for developers to build Blockchains based on Tendermint consensus conveniently such that more blockchains can join the system and gradually extend the scalability of a network. With multiple parallel chains running in the network, Cosmos can achieve a horizontal scalability.

Unfortunately, the popular PoW-featured blockchain such as Bitcoin and Ethereum, cannot connect to Cosmos Hub directly. An alternative solution is to create a customized Peg-zone (like Pegged chain shown in Figure 10) as a bridge to exchange data.

b: POLKADOT

Polkadot [31] also outlined a multi-chain protocol that provides a *relay-chain* to connect heterogeneous blockchains. As mentioned already, relay-chain enables an independent blockchain, an example which is called *parachain* in Polkadot, to exchange information and trust-free inter-chain transactability. In addition, parachain bridge can link to already running blockchains such as Ethereum.

All these proposals employed are able to achieve interoperability and scalability.

IV. FUTURE DIRECTIONS AND OPEN ISSUES

Section III introduces many solutions proposed in recent years dedicated to solving the scalability of blockchain. However, there is still no method that can be applied to existing well-known blockchain systems and solve this problem perfectly. To this end, we should continue to explore and improve existing solutions to achieve a better effect. Here are a few possible directions.

A. LAYER-1

Layer-1 solutions have been studied widely, but it still requires more explorations for scalability solutions. We envision open issues in the directions of *block data* and *sharding techniques*.

1) BLOCK DATA

Despite other methods concerning scalability, the individual nodes' limited capability of storage and bandwidth will be the performance bottleneck of blockchain systems. Firstly, increasing TPS indicates that much more block data need to be propagated within the system, which may aggravate the congestion problems. Besides, as the blockchain grows, more blockchain data should be stored by individual nodes. It will increase the pressure of storage and promote the tendency to centralization. Many discussions about chain pruning [110]–[112] have been proposed. Blockchain pruning approaches aim to remove some historical data that is not critical from the blockchain while preserving the security. The reduction of data releases the storage pressure of full nodes in the blockchain. Therefore, to keep developing blockchain, solutions related to block compression and blockchain data pruning require more optimization and should be applied to real blockchain systems.

2) SHARDING TECHNIQUES

The sharding technique is a popular and effective solution. A sharding-based blockchain is divided into different shards with proper mechanisms to manage each shard as well as transactions and scales horizontally with the number of nodes. However, the following two issues are still open for further investigations:

(1) How to place transactions into different shards. 95% transactions in OmniLedger [12] are cross-shard transactions, leading to much bandwidth pressure because of the communication cost of cross-shard transactions and thus decrease the performance of the whole sharding-based system. Besides the communication cost, reconfiguring shards also cause the exchange of a great amount of data. Therefore, better algorithms should be provided to solve the problem.

(2) How to improve the efficiency of cross-shard transactions. The existing solutions have achieved several good results by their cross-shard submission protocols. However, since cross-shard transactions involve multiple shards and lead to more bandwidth consumption and longer confirmation time, a more efficient protocol is needed to reduce the confirmation latency. This direction still has a large room to explore.

B. LAYER-2

Regarding the Layer-2 solutions, some of them are still in their work-in-progress stages. In particular, Lightning Network is under the spotlight. Many teams have developed the Lightning Network clients and have achieved a high user-of-experience through a series of improvements in the

routing mechanism. When the Ethereum's Plasma framework was proposed, many follow-up teams implemented it to varying degrees, proving the high recognition of sidechain technology. According to the prototypes outlined in this paper, the subsequent studies should focus on the relationship between the sidechain and the main chain, and how to scale out the blockchain and achieve substantial improvement on overall performance while ensuring its fundamental properties. Cross-chain solutions, like Cosmos and Polkadot, have devised their dedicated protocols in order to build a network of heterogeneous blockchain.

C. LAYER-0

We particularly review some new solutions proposed recently and classify them into the category of *Layer-0*. This type of solutions concern the optimizations of the dissemination protocol for information (transaction or block) in the blockchain network. Nodes in the blockchain network broadcast blocks and transactions to the network, but the broadcast is not efficient enough, leading to latency and high bandwidth usage. Some solutions related to block compression discussed above like *Compact Blocks* [10], also focus on the optimization of block propagation, and thus can be viewed as a Layer-0 solution. As mentioned before, faster block propagation leads to larger blocks and shorter block intervals, thereby increasing transaction throughput. Thus, the protocols aiming at optimizing the data propagation in blockchains are desired in future scalable blockchain systems.

Several approaches intending to improve the propagation protocol have been proposed. For instance, Erlay [46] optimizes Bitcoin's transaction relay protocol to reduce the overall bandwidth consumption while increasing the propagation latency. Velocity [48] also brings some improvement in block propagation by utilizing Fountain code, a kind of erasure code, to reduce the amount of data be propagated. Kadcast [47] proposes an efficient block propagation approach based on overlay structure of Kademlia [113]. bloXroute [49] is a Blockchain Distribution Network (BDN) that helps individual nodes to propagate transactions and blocks more quickly. Besides these solutions, there remains a lot of room for optimizations of propagation protocols of current blockchain systems, such as better routing mechanisms, that will contribute to the improvement of the scalability of blockchain.

V. CONCLUSION

Blockchain technologies have grown rapidly in the past few years and will be applied to more applications in different fields in the foreseeable near future. With the increasing adoption of blockchain technology, the number of users has steadily increased. However, the network congestion problem that has occurred many times and enforced people to carefully think about how to solve the scalability issue of blockchains. To this end, a number of new solutions have been proposed. In this paper, we describe the blockchain performance problem mainly paying attention to scalability, and then classify the existing mainstream solutions into several representative layers. Besides, we elaborate some popular solutions such as Sharding, Sidechain, and cross-chain, intending to give a comprehensive explanation. Furthermore, we also summarize several potential research directions and open issues based on the drawback found, such as the huge amount of blockchain data that need to be compressed or pruned, the inefficient cross-shard transaction and unfinished protocols to bridge the existing blockchain to cross-chain platforms, aiming at addressing the scalability of blockchain systems.

By this comprehensive survey, we expect our classification and the analysis over the current solutions can inspire further booming studies dedicated to improving the scalability of blockchains.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2008.
[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.
[3] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, vol. 310, 2016.
[4] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, p. 352, 2018.
[5] P. Zheng, Z. Zheng, X. Luo, X. Chen, and X. Liu, "A detailed and real-time performance monitoring framework for blockchain systems," in *Proc. 40th Int. Conf. Softw. Eng. Softw. Eng. Pract.-ICSE-SEIP*, 2018, pp. 134–143.
[6] *The Scalability Trilemma in Blockchain*. Accessed: Sep. 1, 2019. [Online]. Available: https://medium.com/@aakash_13214/the-scalability-trilemma-in-blockchain-75fb57f646df
[7] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *SIGACT News*, vol. 33, no. 2, p. 51, Jun. 2002.
[8] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, to be published.
[9] *Bitcoin Cash*. Accessed: Sep. 1, 2019. [Online]. Available: https://www.bitcoincash.org/
[10] *Bip152*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0152.mediawiki
[11] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.-CCS*, 2016, pp. 17–30.
[12] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.
[13] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Jan. 2018, pp. 931–948.
[14] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 95–112.
[15] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2016, pp. 45–59.
[16] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Operating Syst. Princ.-SOSP*, 2017, pp. 51–68.
[17] I. Bentov, R. Pass, and E. Shi, "Snow white: Provably secure proofs of stake," *IACR Cryptol. ePrint Archive*, vol. 2016, p. 919, Sep. 2016.
[18] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Proc. Annu. Int. Cryptol. Conf.* Santa Barbara, CA, USA: Springer, 2017, pp. 357–388.
[19] B. David, P. Gaži, A. Kiayias, and A. Russell, "Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Tel Aviv, Israel: Springer, 2018, pp. 66–98.
[20] J. Poon and T. Dryja. (2016). *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments*. [Online]. Available: https://www.bitcoinlightning.com

[21] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," White Paper, 2017, pp. 1–47. [Online]. Available: https://www.plasma.io
[22] Cosmos. Accessed: Sep. 1, 2019. [Online]. Available: https://cosmos.network/whitepaper
[23] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, D. Song, R. Wattenhofer, and E. G. Sirer, "On scaling decentralized blockchains," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Christ Church, Barbados: Springer, 2016, pp. 106–125.
[24] *Scalability of Bitcoin*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Scalability
[25] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proc. 25th USENIX Security Symp. USENIX Secur.*, 2016, pp. 279–296.
[26] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Proc. Symp. Self-Stabilizing Syst.* Edmonton, AB, Canada: Springer, 2015, pp. 3–18.
[27] *Raiden Network*. Accessed: Sep. 1, 2019. [Online]. Available: https://raiden.network/
[28] A. Miller, I. Bentov, S. Bakshi, R. Kumaresan, and P. McCorry, "Sprites and state channels: Payment networks that go faster than lightning," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Frigate Bay, St. Kitts: Springer, 2019, pp. 508–526.
[29] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille. (2014). *Enabling Blockchain Innovations With Pegged Sidechains*. [Online]. Available: http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains
[30] *Liquidity Network*. Accessed: Sep. 1, 2019. [Online]. Available: https://liquidity.network/
[31] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," Polkadot, White Paper, 2016.
[32] J. Teutsch and C. Reitwießner, "A scalable verification solution for blockchains," 2019, *arXiv:1908.04756*. [Online]. Available: https://arxiv.org/abs/1908.04756
[33] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *Proc. 27th USENIX Secur. Symp. USENIX Secur.*, 2018, pp. 1353–1370.
[34] E. Lombrozo, J. Lau, and P. Wuille, "Segregated witness (consensus layer)," Bitcoin Core Develop. Team, Tech. Rep., 2015.
[35] D. Ding, X. Jiang, J. Wang, H. Wang, X. Zhang, and Y. Sun, "Txilm: Lossy block compression with salted short hashing," 2019, *arXiv:1906.06500*. [Online]. Available: https://arxiv.org/abs/1906.06500
[36] Z. Xu, S. Han, and L. Chen, "CUB, a consensus unit-based storage scheme for blockchain system," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 173–184.
[37] X. Dai, J. Xiao, W. Yang, C. Wang, and H. Jin, "Jidar: A jigsaw-like data reduction approach without trust assumptions for bitcoin system," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1317–1326.
[38] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* San Juan, Puerto Rico: Springer, 2015, pp. 528–547.
[39] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," *IACR Cryptol. ePrint Archive*, vol. 2016, p. 1159, 2016.
[40] Y. Sompolinsky and A. Zohar, "Phantom: A scalable blockdag protocol," *IACR Cryptol. ePrint Archive*, vol. 2018, p. 104, 2018.
[41] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," 2018, *arXiv:1805.03870*. [Online]. Available: https://arxiv.org/abs/1805.03870
[42] S. D. Lerner. (2015). *Dagcoin: A Cryptocurrency Without Blocks*. [Online]. Available: https://bitslog.com/2015/09/11/dagcoin/
[43] *Iota*. Accessed: Sep. 1, 2019. [Online]. Available: https://www.iota.org/
[44] A. Churyumov. (2016). *Byteball: A Decentralized System For Storage and Transfer of Value*. [Online]. Available: https://byteball.org/Byteball.pdf
[45] C. LeMahieu. *Nano: A Feeless Distributed Cryptocurrency Network*. Accessed: Mar. 24, 2018. [Online]. Available: https://nano.org/en/whitepaper
[46] G. Naumenko, G. Maxwell, P. Wuille, S. Fedorova, and I. Beschastnikh, "Bandwidth-efficient transaction relay for bitcoin," 2019, *arXiv:1905.10518*. [Online]. Available: https://arxiv.org/abs/1905.10518
[47] E. Rohrer and F. Tschorsch, "Kadcast: A structured approach to broadcast in blockchain networks," in *Proc. 1st ACM Conf. Adv. Financial Technol.*, 2019, pp. 199–213.
[48] N. Chawla, H. W. Behrens, D. Tapp, D. Boscovic, and K. S. Candan, "Velocity: Scalability improvements in block propagation through rateless erasure coding," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 447–454.
[49] U. Klarman, S. Basu, A. Kuzmanovic, and E. G. Sirer, "Bloxroute: A scalable trustless blockchain distribution network whitepaper," White Paper.
[50] I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A. B. Tran, and P. Rimba, "On availability for blockchain-based systems," in *Proc. IEEE 36th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2017, pp. 64–73.
[51] *Decentralized Application*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Decentralized_application/
[52] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion Environ. Sustainability*, vol. 28, pp. 1–9, Oct. 2017.
[53] *Decentralized Application*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
[54] *First Bitcoin Cash Block Mined*. Accessed: Sep. 1, 2019. [Online]. Available: https://news.bitcoin.com/fork-watch-first-bitcoin-cash-block-mined/
[55] *Block size limit controversy*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Block-size-limit_controversy
[56] *Block Size Increase*. Accessed: Sep. 1, 2019. [Online]. Available: https://bitfury.com/content/downloads/block-size-1.1.1.pdf
[57] S. Elmohamed. *Towards Massive On-Chain Scaling: Block Propagation Results With Xthin*. Accessed: Sep. 1, 2019. [Online]. Available: https://medium.com/@peter_r/towards-massive-on-chain-scaling-block-propagation-results-with-xthin-a0f1e3c23919
[58] *Lumino Transaction Compression Protocol(LTCP)*. Accessed: Sep. 1, 2019. [Online]. Available: https://docs.rsk.co/LuminoTransactionCompressionProtocolLTCP.pdf
[59] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* San Juan, Puerto Rico: Springer, 2015, pp. 507–527.
[60] *Casper-Proof-of-Stake-Compendium*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/Casper-Proof-of-Stake-compendium
[61] D. Larimer, "Delegated proof-of-stake (dpos)," Bitshare, White Paper, 2014.
[62] *Bitshares Blockchain*. Accessed: Sep. 1, 2019. [Online]. Available: https://bitshares.org/
[63] *eosio, The Most Powerful Infrastructure for Decentralized Applications*. Accessed: Sep. 1, 2019. [Online]. Available: https://eos.io/
[64] *Eos:Less Than 1% of EOS Addresses Hold 86% of the Tokens!* Accessed: Sep. 1, 2019. [Online]. Available: https://medium.com/@freetokencryptobounty/eos-less-than-1-of-eos-addresses-hold-86-of-the-tokens-5ad4b2eac403
[65] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. OSDI*, vol. 99, 1999, pp. 173–186.
[66] *Byzantine Fault*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Byzantine_fault
[67] R. Canetti and T. Rabin, "Fast asynchronous Byzantine agreement with optimal resilience," in *Proc. 25th Annu. ACM Symp. Theory Comput.-STOC*, 1993, pp. 42–51.
[68] D. Malkhi and M. Reiter, "Unreliable intrusion detection in distributed computations," in *Proc. 10th Comput. Secur. Found. Workshop*, Nov. 2002, pp. 116–124.
[69] J. A. Garay and Y. Moses, "Fully polynomial byzantine agreement for n > 3t processors in t + 1 rounds," *SIAM J. Comput.*, vol. 27, no. 1, pp. 247–290, Feb. 1998.
[70] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, Tendermint, 2016.
[71] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford, "Keeping authorities 'honest or bust' with decentralized witness cosigning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 526–545.
[72] R. Pass and E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model," in *Proc. 31st Int. Symp. Distrib. Comput. (DISC)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
[73] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, "Solida: A blockchain protocol based on reconfigurable byzantine consensus," 2016, *arXiv:1612.02916*. [Online]. Available: https://arxiv.org/abs/1612.02916
[74] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, Jan. 2003, , pp. 120–130.
[75] *Proof of Authority-Wikipedia*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Proof_of_authority
[76] *Proof-of-Capacity*. Accessed: Sep. 1, 2019. [Online]. Available: https://burstwiki.org/en/proof-of-capacity/

[77] A. Nandwani, M. Gupta, and N. Thakur, "Proof-of-participation: Implementation of proof-of-stake through proof-of-work," in *Proc. Int. Conf. Innov. Comput. Commun.* New Delhi, India: Springer, 2019, pp. 17–24.
[78] *Shard Wiki*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Shard_(database_architecture
[79] E. Fynn and F. Pedone, "Challenges and pitfalls of partitioning blockchains," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2018, pp. 128–133.
[80] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Tech. J.*, vol. 49, no. 2, pp. 291–307, Feb. 1970.
[81] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Jan. 1998.
[82] E. Syta, P. Jovanovic, E. K. Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable bias-resistant distributed randomness," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 444–460.
[83] *Light-Weight User*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.bitcoin.it/wiki/Lightweight_node
[84] *The Zilliqa Technical Whitepaper*, Z. Team, Oakbrook Terrace, IL, USA, Sep. 2017, vol. 16, p. 2019.
[85] *The Harmony Team. Open Consensus for 10 Billion People*. Accessed: Sep. 1, 2019. [Online]. Available: https://harmony.one/
[86] *Dag(Directed Acyclic Graph)*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Directed_acyclic_graph
[87] *Fantom*. Accessed: Sep. 1, 2019. [Online]. Available: https://fantom.foundation/
[88] S. Popov, "The tangle," *cit. on*, p. 131, 2016.
[89] B. Kusmierz, "The first glance at the simulation of the tangle: Discrete model," IOTA, Tech. Rep., 2017.
[90] B. Kusmierz, P. Staupe, and A. Gal, "Extracting tangle properties in continuous time via large-scale simulations," Tech. Rep., 2018.
[91] A. Cullen, P. Ferraro, C. King, and R. Shorten, "Distributed ledger technology for iot: Parasite chain attacks," 2019, *arXiv:1904.00996*. [Online]. Available: https://arxiv.org/abs/1904.00996
[92] L. Baird, M. Harmon, and P. Madsen, "Hedera: A governing council & public hashgraph network," *Trust layer Internet, Whitepaper*, vol. 1, 2018.
[93] T. Rocket, "Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies," IPFS, Tech. Rep., 2018.
[94] *Complaints About IOTA*. Accessed: Sep. 1, 2019. [Online]. Available: https://juejin.im/post/5c6e0f0bf265da2de66103dd
[95] *Double-Spending*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Double-spending
[96] *Lightning Labs*. Accessed: Sep. 1, 2019. [Online]. Available: https://lightning.engineering/
[97] *Bitmex-The Lightning Network*. Accessed: Sep. 1, 2019. [Online]. Available: https://blog.bitmex.com/the-lightning-network/
[98] *Erc20 Token Standard*. Accessed: Sep. 1, 2019. [Online]. Available: https://theethereum.wiki/w/index.php/ERC20_Token_Standard
[99] *Lightning Network Daemon*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/lightningnetwork/lnd
[100] *C-Lightning—A Lightning Network Implementation in C*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/ElementsProject/lightning
[101] *A Scala Implementation of the Lightning Network*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/ACINQ/eclair
[102] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019.
[103] *Simplified Payment Verification*. Accessed: Sep. 1, 2019. [Online]. Available: https://en.bitcoinwiki.org/wiki/Simplified_Payment_Verification
[104] *Minimal Viable Plasma*. Accessed: Sep. 1, 2019. [Online]. Available: https://ethresear.ch/t/minimal-viable-plasma/426
[105] *Minimal Viable Plasma*. Accessed: Sep. 1, 2019. [Online]. Available: https://ethresear.ch/t/plasma-cash-plasma-with-much-less-per-user-data-checking/1298
[106] *Sparse Merkle Trees*. Accessed: Sep. 1, 2019. [Online]. Available: https://ethresear.ch/t/optimizing-sparse-merkle-trees/3751
[107] *Plasma Debit: Arbitrary-Denomination Payments in Plasma Cash*. Accessed: Sep. 1, 2019. [Online]. Available: https://ethresear.ch/t/plasma-debit-arbitrary-denomination-payments-in-plasma-cash/2198
[108] R. Khalil and A. Gervais, "NOCUST–a non-custodial 2nd-layer financial intermediary," Cryptology ePrint Archive, Report 2018/642, 2018. [Online]. Available: https://eprint.iacr.org/2018/642
[109] *Tendermint*. Accessed: Sep. 1, 2019. [Online]. Available: https://tendermint.com/
[110] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain mutability: Challenges and proposed solutions," 2019, *arXiv:1907.07099*. [Online]. Available: https://arxiv.org/abs/1907.07099
[111] M. Florian, S. Henningsen, S. Beaucamp, and B. Scheuermann, "Erasing data from blockchain nodes," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Jun. 2019, pp. 367–376.
[112] *Ethereum Chain Pruning for Long Term 1.0 Scalability and Viability*. Accessed: Sep. 1, 2019. [Online]. Available: https://ethereum-magicians.org/t/ethereum-chain-pruning-for-long-term-1-0-scalability-and-viability/2074
[113] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Int. Workshop Peer-to-Peer Syst.* Cambridge, MA, USA: Springer, 2002, pp. 53–65.

QIHENG ZHOU is currently pursuing the bachelor's degree with the School of Data and Computer Science, Sun Yat-sen University, China. His current research interest is blockchain.

HUAWEI HUANG (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Aizu, Japan, in 2016. He is currently an Associate Professor with Sun Yat-sen University, China. His research interests include blockchain and intelligent computing. He has served as a Research Fellow of JSPS (2016–2018); a Visiting Scholar of The Hong Kong Polytechnic University (2017–2018); and an Assistant Professor with Kyoto University, Japan (2018–2019). He is a member of the ACM. He received the Best Paper Award from TrustCom2016.

ZIBIN ZHENG (Senior Member, IEEE) received the Ph.D. degree from The Chinese University of Hong Kong, in 2011. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, China. He serves as the Chairman of the Software Engineering Department. He has published over 120 international journal and conference papers, including three ESI highly-cited papers. According to Google Scholar, his papers have more than 7000 citations, with an H-index of 42. His research interests include blockchain, services computing, software engineering, and financial big data. He was a recipient of several awards, including the Top 50 Influential Papers in Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at ICSE2010, and the Best Student Paper Award at ICWS2010. He served as BlockSys'19 and CollaborateCom'16 General Co-Chair, SC2'19, ICIOT'18, and the IoV'14 PC Co-Chair.

JING BIAN received the B.Sc. degree in automation, the M.Sc. degree in computational mathematics, and the Ph.D. degree in physics from Sun Yat-sen University, Guangzhou, China, in 1988, 2001, and 2006, respectively. She is currently a Vice-Professor with the School of Data and Computer Science, Sun Yat-sen University. Her current research interests include design and analysis of algorithms, blockchain, electronic commerce, and social networks.