# A Survey on Blockchain Scalability Technologies

Rui Chang[1], Xuewen Dong[1], Wei Tong[2], Xiaojie Guo[1], Zesong Dong[1], and Yiheng Lou[1]
[1]College of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071, China
[2]School of Information Science and Engineering, Zhejiang University of Science and Technology, Hangzhou, 310018, China

**Blockchain technology, with its features of decentralization, immutability, and traceability, shows broad application potential in fields such as finance, supply chain, and the Internet of Things (IoT). However, as application scenarios expand and the number of users surges, blockchain systems face severe challenges in scalability. Based on the limitations of the traditional single-chain architecture, this paper comprehensively reviews mainstream blockchain scaling solutions from two dimensions: on-chain scaling and off-chain scaling. On-chain scaling covers sharding technology and consensus mechanism optimization; off-chain scaling includes payment channel networks and Layer-2 Rollups. By comparing and analyzing the technical principles, advantages, limitations, and applicable scenarios of various solutions, this paper reveals that "on-chain basic optimization and off-chain collaborative scaling" has become the mainstream direction of blockchain scalability research. Building on this, Directed Acyclic Graphs (DAG) are highlighted in a separate section, detailing the technical principles and representative systems of three DAG solutions: natural structure, parallel structure, and main-chain structure. Finally, the paper looks forward to the future research trends of blockchain data structures and scaling technologies, providing a reference for theoretical research and engineering practice in related fields.**

***Index Terms*—Blockchain, Scalability, Sharding technology, Payment channel networks, Layer-2 Rollups, Directed Acyclic Graph (DAG).**

## I. Introduction

ALTHOUGH blockchain technology has shown great potential in establishing trust mechanisms in many fields, its scalability bottleneck is seriously limiting the progress of its large-scale commercial application. Since the release of the Bitcoin white paper in 2008, blockchain technology has evolved from the underlying technology of a single cryptocurrency to an important infrastructure supporting the digital economy [1]. With core features such as decentralization, immutability, and traceability, blockchain technology has shown huge application potential in financial services, supply chain management, digital identity authentication, and IoT collaboration, providing a new technical path for rebuilding trust mechanisms in traditional business processes [2]. However, with the continuous expansion of application scenarios and the rapid growth of users, the scalability bottleneck of blockchain systems has become increasingly obvious [3]. The conflict between transaction processing capacity and market demand has become more prominent, severely limiting its large-scale commercial application.

Since traditional blockchains use a network-wide consensus mechanism, transaction processing efficiency has long been at a low level, making it difficult to support application demands like high-frequency trading, micropayments, and large-scale real-time interactions in IoT. Traditional blockchain [4] systems generally use a network-wide consensus mechanism, requiring each transaction to be broadcast to all network nodes and agreed upon before it can be confirmed on the chain. Although this design effectively ensures the security and consistency of the system and prevents malicious behaviors like double spending, it keeps transaction processing efficiency low. Taking mainstream public chains as an example, the Bitcoin network, limited by block size and block interval, has a maximum sustainable throughput of about 7 transactions per second (TPS); before completing its transition to Proof of Stake, Ethereum long maintained about 15 TPS due to the single-block Gas limit and global state update mechanism [5]. This significant performance gap makes it difficult for current blockchain systems to effectively support high-frequency trading scenarios (like stock exchanges), micropayment applications (like pay-per-view content), and large-scale real-time interactions between IoT devices that require high throughput and low response latency.

From a technical root perspective, the scalability bottleneck of blockchains is closely related to their basic data structure design [6]. Traditional blockchains use a single-chain data structure, where blocks must be generated and submitted strictly according to a global time sequence, forming a linear block connection structure. Although this serial organization helps preserving the consistency of the global state and the verifiability of transaction history, it also fundamentally limits the concurrent processing capacity of the system. Nodes cannot generate and confirm multiple blocks in parallel, which limits the potential for improving overall throughput performance [7]. In addition, the design of the consensus mechanism also directly restricts the network scalability of the system: the communication complexity of voting-based Byzantine Fault Tolerance consensus protocols (such as PBFT and its variants) grows quadratically with the number of nodes, resulting in limited network capacity; although Proof of Work consensus protocols can support more nodes, they face challenges such as high energy consumption, long confirmation delays, and concentrated computing power [8].

To break through the scalability bottleneck, academia and industry have proposed two main types of solutions for linear chains: on-chain optimization and off-chain scaling [9]. On-chain optimization improves the system's processing capacity without changing the location where transactions are executed, mainly through sharding and optimizing consensus mech-

Manuscript received May 6, 2026; revised August 17, 2026. Corresponding author: Xuewen Dong (email: xwdong@xidian.edu.cn).

anisms. Typical solutions include sharding technology and multi-round improvements of consensus algorithms. Off-chain scaling transfers a large number of transactions to be executed off the main chain, only interacting with the main chain at necessary points such as opening a transaction, state commitment, or dispute arbitration. Representative technologies include payment channel networks and state channels [10]. To provide a strong problem-driven narrative, it is essential to recognize that as smart contracts and Decentralized Finance (DeFi) ecosystems explode in complexity, the demand for scalable infrastructure has outpaced early off-chain solutions. These two types of solutions work together through layered design to build a multi-level blockchain scaling system to meet the different needs for security, decentralization, and throughput in various application scenarios. To break the throughput bottleneck caused by the serial processing of transactions in a single-chain structure, researchers introduced Directed Acyclic Graph (DAG) structures into blockchain systems. Unlike linear chains where each block references only one preceding block, the DAG structure allows a block to reference multiple preceding blocks at the same time.

This paper aims to systematically review the scaling methods of blockchains and their evolution paths, comprehensively summarize mainstream blockchain scaling technology solutions, deeply analyze the technical principles, advantages, limitations, and applicable scenarios of various solutions, and on this basis, look forward to possible future research directions. Through the research in this paper, we hope to provide a systematic reference for theoretical research and engineering practice regarding blockchain scalability issues, and promote the large-scale application of blockchain technology in high-value scenarios.

While numerous surveys have explored blockchain scalability, many focus exclusively on isolated dimensions such as sharding or DAGs. Existing literature often lacks a comprehensive, up-to-date synthesis that includes the paradigm shift brought by Layer-2 Rollups (Zero-Knowledge and Optimistic) and modular Data Availability (DA) layers. This survey bridges that critical research gap by providing a holistic taxonomy covering on-chain, off-chain, Layer-2, and DAG technologies up to 2026, comparing their scopes, covered technologies, and methodological perspectives against prior reviews.

To ensure maximum scientific rigor, transparency, and reproducibility, this paper adopts a formalized Systematic Literature Review (SLR) methodology in accordance with the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines. We established strict quantitative inclusion and exclusion criteria: peer-reviewed journal and top-tier conference papers published between 2018 and 2026 that present foundational architectures, empirical evaluations, or routing optimizations for blockchain scaling were included. Non-peer-reviewed whitepapers, outdated preprints, and publications lacking transparent evaluation metrics were excluded. The literature was collected from premier databases including IEEE Xplore, Springer, ACM Digital Library, and Google Scholar using targeted keyword strings (e.g., “blockchain scalability" and (“sharding" or “Rollups" or “payment channel networks" or “DAG")). The step-by-step screening procedure is visually summarized in our PRISMA flowchart.

The structure of this paper is organized as follows: Section 2 reviews on-chain scaling technologies; Section 3 reviews off-chain scaling technologies; Section 4 focuses on DAG-based blockchain structures; Section 5 summarizes the paper and looks forward to future research directions.To address the need for systematic presentation, additional taxonomy figures and comparative tables have been incorporated throughout the sections to visually synthesize the characteristics and application scenarios of these technologies.

## II. On-chain Scaling Technologies

### A. Sharding

Sharding technology was first used in distributed databases to improve data processing performance by having different shards independently and in parallel perform data management operations [11]. In blockchain systems, sharding technology exists as a protocol, mainly modifying the network layer to divide the entire blockchain network into multiple shards. Each shard independently and in parallel processes transactions, thereby improving the overall transaction throughput of the system [12].

#### 1) Challenges of Sharding Technology

Although sharding technology shows huge potential for performance optimization, it still faces a series of technical challenges [13]:

**Shard Load Balancing:** Transaction distribution in real applications often has significant unevenness in time and space. For example, a popular decentralized application (DApp) or asset issuance contract may attract a large number of transactions in a short time, causing its shard to become a performance bottleneck while other shards remain relatively idle. This uneven load reduces the resource utilization and effective throughput of the overall system. To solve this problem, a dynamic load balancing mechanism is needed [14]. This mechanism should be able to redefine shard boundaries (like shard key ranges or account assignments) without affecting system security and internal state consistency, migrating some accounts or states from overloaded shards to idle ones. However, the state migration process itself involves shard data synchronization and temporary double-writing, which is highly complex to implement and may introduce new security vulnerabilities.

**Shard Security:** Sharding technology divides all network nodes into multiple smaller subsets, with each shard running a consensus protocol independently [15]. Compared to a single un-sharded blockchain, sharding reduces the number of nodes in each shard, which may lower the cost for an attacker to control that shard. For example, in a system using random shard allocation, if the total number of nodes is N and the number of shards is K, the average number of nodes per shard is N/K. If an attacker controls more than 1/3 or 1/2 of the malicious nodes in the total nodes (depending on the fault tolerance threshold of the consensus protocol), they have a high probability of controlling a specific shard during a shard reorganization. To defend against such attacks, it is necessary to design random shard allocation and periodic reorganization mechanisms [16], so that the roles and identities of nodes in

shards change dynamically over time, increasing the difficulty for attackers to predict and focus attacks on a specific shard. At the same time, fast detection and penalty mechanisms within shards, as well as global security audits of shards, need to be introduced to ensure that even if a single shard is compromised, it will not affect other shards or corrupt the global state.

**Internal Consensus Efficiency Bottlenecks in Shards:** Although sharding technology improves the system's overall throughput at a macro level by dividing the network into multiple parallel processing subsets, each shard is essentially still an independently running sub-blockchain and must fully execute the consensus protocol internally [17]. This means that the transaction processing capacity of a single shard is still limited by the performance ceiling of its chosen consensus mechanism. For example, if a shard internally uses a Byzantine Fault Tolerance protocol (such as PBFT), its communication complexity grows quadratically with the number of nodes, and a slight increase in nodes will significantly lower confirmation efficiency; if a Proof of Work mechanism is used, it is constrained by block intervals and block sizes, making it difficult to achieve high-frequency confirmations. More importantly, this single-shard performance bottleneck has a "weakest link" effect—when a specific shard experiences a surge in load due to hosting popular applications or massive transactions, that shard itself becomes the performance bottleneck for the entire system [18], and simply increasing the number of shards cannot improve the processing capacity of this specific shard. In other words, the parallelization advantage of sharding is mainly reflected in processing transactions of different shards at staggered times, but for high-value scenarios within a single shard, its throughput ceiling is still determined by the shard's own consensus mechanism and cannot be linearly broken through horizontal scaling.

*2) Main Types of Sharding*

Based on different division dimensions, sharding technology can be divided into three main types: network sharding, transaction sharding, and state sharding. The three types are functionally related and are usually used together in real systems to form a complete sharding architecture.

**Network Sharding:** Network sharding is the foundation of the sharding architecture. It refers to dividing all network nodes into multiple relatively independent sub-networks, with each sub-network forming a shard. The RapidChain system was the first to implement a complete network sharding architecture. By randomly assigning nodes to different shards, it significantly reduced the communication and storage burden on single nodes [19]. Dang et al. pointed out in their research that in traditional single-chain architectures, every node must synchronize the full amount of data and participate in global consensus. As the network scales up, nodes face severe bandwidth and storage pressures [20]. Network sharding groups nodes so that each node only needs to communicate with a few nodes within its shard, thereby reducing the network bandwidth consumption and state storage requirements of a single node . The OmniLedger system uses a verifiable random function to achieve random allocation of nodes, effectively preventing malicious nodes from centrally controlling a specific shard [21].

**Transaction Sharding:** Transaction sharding refers to allocating transactions to different shards for parallel processing based on specific rules [22]. Its core goal is to improve the system's concurrent transaction processing capacity, allowing unrelated transactions to be confirmed simultaneously in different shards. The core advantage of transaction sharding lies in its parallel processing ability: when multiple transactions are assigned to different shards, each shard can independently perform consensus and verification in parallel, and the overall system throughput theoretically grows linearly with the number of shards. However, the main challenge of transaction sharding lies in how to design efficient allocation rules that ensure an even distribution of transaction load among shards while minimizing the generation of cross-shard transactions.

Transaction sharding allocation rules are usually based on key fields in the transaction, such as the sender's address, the receiver's address, or the transaction hash. Common allocation strategies include account address-based sharding and transaction hash-based sharding.

**State Sharding:** State sharding is the most challenging type in sharding technology. It refers to splitting and storing the global state (including account balances, smart contract code, and stored data) across different shards, with each shard maintaining only the state data within its jurisdiction [23]. The fundamental difference between state sharding and traditional sharding architectures is that it is not just a diversion of the transaction processing procedure, but a complete dismantling of the data storage layer. Under the state sharding architecture, each shard has an independent state database, and nodes only need to store the state data of their shard, fundamentally solving the storage bloat problem faced by single-chain architectures [24], enabling the system to support massive users and complex applications.

However, state sharding traditionally faces massive challenges regarding data storage and state migration [25]. To make state sharding viable, recent critical advancements have emerged, namely Data Availability Sampling (DAS) and stateless clients [26]. Data Availability Sampling allows light nodes to cryptographically verify that shard data is available without downloading the entire block, significantly enhancing shard security. Furthermore, stateless clients leveraging Verkle trees [27] compress state proofs, drastically reducing the storage burden on validators and making cross-shard state [28] migrations highly efficient and feasible.

*3) Performance Characteristics of Sharding*

**Parallel Processing Capacity:** The core performance advantage of the sharding architecture lies in its parallel processing capacity [29]. By distributing the transaction load to multiple shards for simultaneous processing, the overall system throughput can theoretically reach the level of "single shard throughput $\times$ number of shards". Experiments in the RapidChain system show that when the number of shards is 8, the system throughput can reach about 7000 TPS, far exceeding the single-chain architecture [30]. However, actual system throughput is often lower than the theoretical maximum, mainly constrained by three factors: first, the shard load balancing issue; if transactions are unevenly distributed,

the most loaded shard will become the overall bottleneck. Second, the single shard performance boundary; the consensus mechanism within each shard still has a throughput limit that cannot be linearly improved by adding shards. Third, transaction dependencies; transactions with dependencies must be processed serially in the same shard, which reduces parallelism.

**Resource Decentralization and Storage Optimization:** Xiangdong Guo et al. designed a jump-hash sharding algorithm to optimize data allocation, and combined it with an enhanced consensus algorithm to improve consensus efficiency, significantly improving the performance and scalability of an agricultural traceability blockchain system based on sharding technology. The sharding architecture significantly lowers the barrier to joining the network by dispersing resource requirements, which helps improve the network's degree of decentralization [31]. Yan Wang et al. proposed that in terms of storage resources, state sharding allows each node to store only the state data of its shard, rather than the global state. Assuming the total global state is $S$ and the number of shards is $K$, the theoretical average storage requirement per node is about $S/K$. This allows ordinary hardware devices to participate in network verification [32], avoiding centralization tendencies caused by high storage requirements.

**Performance Trade-offs and Limitations:** While improving scalability, sharding technology faces multiple performance trade-offs and limitations [33]. In terms of security, random allocation and periodic reorganization consume resources, which may affect continuous availability, and higher security requirements lead to greater performance loss. Regarding the single-shard bottleneck, each shard is still limited by its internal consensus mechanism, and cannot break through by adding more shards during load surges. In terms of storage, state sharding only disperses rather than solves the continuous swelling of total state data. Finally, regarding node heterogeneity, low-performance nodes may slow down shard efficiency [34].

### *B. Consensus Mechanism Optimization*

The consensus protocol is the core of distributed blockchain systems for achieving state consistency, and its design directly affects the network scalability and performance scalability of the system [35]. Depending on the method of reaching consensus, existing consensus protocols are divided into two categories: proof-based consensus protocols and voting-based consensus protocols [36]. Table I compares various on-chain scaling mechanisms from multiple dimensions.

#### *1) Proof-based Consensus Protocols*

Proof-based consensus protocols (such as Proof of Work PoW, Proof of Stake PoS, etc.) require the winner node to broadcast a proof message to all consensus participant nodes. Participant nodes determine whether to accept the new block and reach a consensus by verifying the correctness of the proof message [37]. The core feature of this type of protocol is that reaching a consensus does not rely on multiple rounds of complex voting interactions, but on a single winner presenting publicly verifiable cryptographic evidence. Other nodes only need to perform a single verification to make a judgment. Therefore, its communication complexity is significantly lower than that of Byzantine Fault Tolerance protocols——usually only $O(w)$, where $w$ is the number of winner nodes. This allows it to support more network nodes participating, thereby maintaining a high degree of decentralization. However, this type of protocol also faces inherent challenges: on the one hand, generating a winner usually requires consuming massive computational resources (like hash operations in PoW) or locking high-value assets (like staked tokens in PoS), leading to significant energy consumption or capital costs; on the other hand, proof-based consensus often uses probabilistic finality, and transaction confirmation requires waiting for the cumulative confirmation of multiple blocks, resulting in long confirmation delays that are difficult to meet the needs of high-frequency, low-latency scenarios.

The Proof of Work (PoW) consensus protocol was first applied to the Bitcoin blockchain. Its core mechanism requires all consensus participant nodes to independently solve the same mathematical puzzle (i.e., finding a random number, nonce, that meets the target difficulty) [1]. The node that solves the puzzle first becomes the winner of the round and broadcasts its packaged block and answer (nonce) to all network nodes. After receiving the broadcast, other nodes only need to perform a hash verification on the answer to determine if the block is valid, thereby quickly reaching consensus [38]. The communication complexity of PoW is $O(w)$, where $w$ is the number of winner nodes (in typical PoW, there is only one winner per round, so the communication overhead is linear with the number of network nodes, i.e., each winner broadcasts once to all nodes).

The Proof of Stake (PoS) consensus protocol is an alternative solution proposed to solve the energy consumption problem of PoW. Its core design concept is: only nodes that own a certain number of tokens (i.e., stakes) can become true consensus participants (validators). The probability of a node gaining block-producing rights or voting rights is directly proportional to the number of tokens it holds, without the need for massive meaningless hash calculations [39]. In typical PoS implementations, the system selects the block-producing node for the next round through a random algorithm (such as one based on a Verifiable Random Function, VRF). The winner broadcasts its block and proof of stake (like a signed random number), and other nodes accept the block after verifying the signature and the validity of the stake. The communication complexity of PoS is also $O(w)$, and is independent of the network scale, so it can theoretically support thousands of validators. Compared to PoW, PoS drastically reduces energy consumption [40], but the confirmation delay problem still exists: many PoS protocols use voting-based finality components (like Casper FFG in Ethereum Gasper), which makes final transaction confirmation still take tens of seconds to several minutes. In addition, PoS also faces unique challenges such as the "Nothing-at-Stake" attack and long-range attacks, which require extra mechanisms (like slashing conditions and weak subjectivity checkpoints) to prevent. Nevertheless, thanks to its high energy efficiency and good scalability, PoS has become the mainstream consensus choice for the current new

generation of public chains (like Ethereum 2.0, Cardano, and Polkadot).

*2) Voting-based Consensus Protocols*

Voting-based consensus protocols reach a consistent consensus through negotiation and multi-round voting among nodes. They can provide strong consistency guarantees (meaning that once a transaction is confirmed, it is irreversible and there is no risk of forking), but the communication complexity is high and network capacity is limited [41]. This type of protocol usually requires all participating nodes to exchange a large number of voting messages in each round of consensus. As the number of nodes increases, the network bandwidth and latency overhead grow rapidly, thereby limiting the number of nodes the system can support.

The Practical Byzantine Fault Tolerance (PBFT) consensus protocol is a classic and representative Byzantine fault tolerance consensus protocol. Its consensus process contains 5 phases: request (the client initiates a request), pre-prepare (the primary node broadcasts a pre-prepare message), prepare (replica nodes broadcast prepare messages), commit (replica nodes broadcast commit messages), and reply (replica nodes reply to the client) [42]. PBFT requires all consensus participants to frequently broadcast voting messages to reach consistency. Its communication complexity is $O(N^2)$, meaning the total number of messages in the network is proportional to the square of the number of nodes. In real network environments, when the number of nodes increases, network bandwidth and CPU overhead will rise sharply. Tests have shown that the network capacity of the PBFT protocol on a 100 Mbps network does not exceed 20 nodes, meaning it can only support about 20 consensus nodes, which severely limits its application in large consortium chains or public chains [5].

HotStuff is a BFT consensus protocol with linear communication complexity. By introducing threshold signatures and pipeline design, it reduces the communication complexity from $O(N^2)$ to $O(N)$ [43]. Threshold signatures allow a group of nodes to jointly generate an aggregated signature, compressing multi-round voting messages into a single message; pipeline design allows consensus operations in different stages to be executed in an overlapping manner, further reducing confirmation delay. HotStuff has good engineering practicality and was used in the Libra blockchain project (later renamed Diem) led by Facebook (now Meta), showing excellent performance in consortium chain scenarios.

*3) Hybrid Consensus Protocols*

Hybrid consensus protocols combine two or more existing consensus protocols to leverage their respective advantages in security, scalability, finality, and energy efficiency, compensating for the inherent defects of a single protocol [44]. These protocols usually divide the consensus process into different stages (like a block generation stage and a confirmation stage) or use different consensus mechanisms at different network layers to achieve overall performance optimization. For example, the PBFT-PoW consensus protocol is a hybrid protocol that combines PBFT and PoW, applied in the TrueChain project [45]. In this protocol, the PoW mechanism is responsible for electing block proposing nodes, ensuring openness and the degree of network decentralization; while PBFT is used for fast voting confirmation on generated blocks, providing strong consistency and finality. This division of labor allows TrueChain to utilize PoW to defend against Sybil attacks while leveraging PBFT to achieve low-latency transaction confirmation, thereby alleviating the problems of long confirmation times and low throughput in traditional PoW. Another typical example is the BFT-DPoS consensus protocol, which combines BFT and DPoS (Delegated Proof of Stake), significantly improving the confirmation speed of the blockchain by combining their features [46]. Its workflow is usually divided into two steps: first, the DPoS protocol is executed, where token-holding voters elect a limited number of delegate nodes (supernodes) that take turns generating blocks in sequence, ensuring high efficiency and predictability in block generation; subsequently, for the newly generated block, the BFT protocol (like PBFT or its variants) is immediately executed, and the same group of delegate nodes conducts multi-round voting to quickly confirm the block, allowing the block to reach a final confirmed state before the next block is generated. This hybrid design of "DPoS block generation + BFT confirmation" not only avoids the risk of forking or long-range attacks that may occur in DPoS, but also overcomes the limitation on node scale in traditional BFT protocols, and is widely used in projects like EOS and Cosmos, achieving second-level transaction finality.

## III. Off-chain Scaling Technologies

Payment Channel Networks (PCN) significantly reduce the load on the main chain by moving high-frequency, small-value transactions off-chain to execute, relying on the blockchain for state confirmation only when channels are opened and closed [47]. Typical implementations, such as the Lightning Network on Bitcoin and the Raiden Network on Ethereum, are based on smart contracts to build bidirectional payment channels, allowing both trading parties to conduct countless instant, zero-fee transactions within the channel, and ultimately only submitting the net settlement state to the main chain [48].

### A. Working Principle of Payment Channels

The workflow of a payment channel includes the following three stages [49], as shown in the original text's Figure 1:

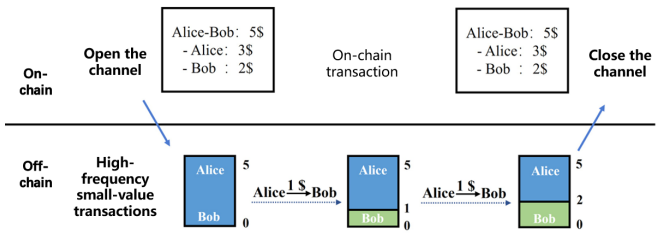



Fig. 1. The process of off-chain channel transactions

**Initialization:** Both parties lock funds in a multi-signature smart contract. For example, Alice and Bob deploy a contract on the chain and deposit 5 BTC.

TABLE I
COMPARATIVE SUMMARY OF ON-CHAIN SCALING MECHANISMS

| Scaling Paradigm | Core Target | Complexity | TPS | Primary Trade-off |
|---|---|---|---|---|
| Network/Tx Sharding | Bandwidth & CPU | $O(N/K)$ | 1,000 – 7,000 | Cross-shard overhead; Takeover risk |
| State Sharding (DAS+Verkle) | Storage bloat | $O(1)$ | 5,000 – 10,000+ | Proof generation cost |
| Voting BFT (HotStuff) | Instant finality | $O(N)$ | 1,000 – 3,000 | Strict node limits |
| Hybrid BFT-DPoS | Fast confirmation | $O(M)$ | 3,000 – 8,000+ | Centralization risk |

**Off-chain Transaction:** Both parties update the balance state through signatures without needing on-chain operations. For example, when Alice pays Bob 1 BTC, both parties sign the new state (Alice: 4 BTC, Bob: 1 BTC).

**Closing:** The final state is submitted to the chain for settlement. When the channel closes, the on-chain balance is updated (Alice: 3 BTC, Bob: 2 BTC).

The payment channel mechanism effectively eliminates on-chain confirmation delays and miner fees by migrating transaction processing off-chain [50]. Taking the Bitcoin network as an example, the on-chain transaction throughput is about 7 TPS, whereas payment channels can process transactions near-instantly.

### *B. Payment Channel Networks and Multi-hop Routing*

In a PCN, when there is no direct payment channel between the payer and the payee, the transaction needs to be relayed and forwarded through several intermediate nodes. This process is called multi-hop routing [51]. In large PCNs, efficient path discovery and selection from the payer to the receiver is a complex task. Routing algorithms need to balance multiple objectives such as path length, fee costs, payment success rates, and privacy protection [52].

In 2024, a SoK (Systematization of Knowledge) published by Kolachala et al. at IEEE [53] conference reviewed several key aspects of PCN path discovery and routing, including source routing, distributed hash table-based routing, and privacy-preserving routing, comparing the pros and cons of different technical routes. This SoK summarized the unique constraints faced by PCN routing as: feasibility constraints—requiring the balance of every channel on the path to be sufficient to cover the payment amount; timeliness constraints—different users have varying tolerances for cooperation time; and directional cancellation properties—reverse payment transactions can cancel each other out in the channel. These constraints make it difficult to directly migrate and apply routing protocols from traditional ad-hoc networks. Research published by Lobmaier et al. in the “ACM Distributed Ledger Technologies" [54] evaluated the applicability of routing protocols from the field of wireless sensor networks to PCNs. The research first refined the core requirements of PCN routing, selected suitable wireless sensor network routing schemes for adaptation, and conducted experimental evaluations based on metrics such as path length, fees, and payment success rate.

Aiming at the problem of channel balance depletion caused by single-direction multi-hop payments in PCNs, Wei et al. proposed the BAR (Balance-Aware Routing) protocol at IEEE ICC [55], which is a class of distributed balance-aware payment routing schemes. The core insight of BAR is to use the unique property in PCNs—that payments in opposite directions between two users can cancel each other out—to alleviate the channel depletion problem. Wang et al. proposed the AERO routing scheme in the journal “Computer Networks" [56]. AERO introduces a balance coefficient to evaluate the availability of funds within the channel, and combines probability metrics to assess the channel’s transaction capacity, achieving adaptive routing to minimize transaction loss. AERO also introduces a wait-queue scheduling algorithm at transaction nodes, executing transactions only when the channel capacity meets predefined requirements.

In 2025, Corcoran and Spasic proposed an edge device path planning method [57], utilizing service providers to assist in path calculation within PCNs. This work integrates two privacy protection mechanisms: the first generalizes existing blinded paths mechanisms to obscure the start and end points of payment paths; the second obfuscates real requests by generating convincing fake requests. Regarding routing protocols for specific scenarios, Xu et al. proposed the EMCPF (Efficient Multi-party Conditional Payment Flow) protocol [58], a payment protocol compatible with the Lightning Network and featuring privacy protection properties, designed to protect all user identities, reduce locked collateral time, and resist wormhole attacks. EMCPF does not rely on the assumption of ideal anonymous communication channels; instead, it uses layered encryption technology and performs security analysis under the universal composability model, proving that as long as there is at least one honest user in the payment path, no information will be leaked.

### *C. Payment Hubs and Virtual Channels*

Payment channel hubs process off-chain transactions by deploying powerful central nodes in the network, representing an important path to improving PCN efficiency [59]. However, existing Payment Center Hubs (PCH) are often placed in advance without considering the distribution of payment requests, leading to load imbalance issues. A virtual channel is a “bridge" channel built on existing payment channel paths. After a virtual channel is established, intermediate nodes no longer participate in subsequent virtual channel payments, thus releasing intermediate nodes from the burden of frequent relay payments.

Yang et al. proposed Splicer, a highly scalable multi-PCH solution based on Trusted Execution Environments (TEE) [60]. This research transformed the NP-hard PCH placement problem through mixed-integer linear programming, and utilized

supermodular optimization techniques to provide optimal or near-optimal solutions for PCNs of different scales, while achieving load balancing. Based on the same research, Yang et al. further proposed the SHARE scheme [61], expanding from a single-hub to a multi-hub distributed routing architecture. SHARE uses a TEE-assisted, privacy-preserving distributed routing protocol to dynamically adjust multi-path traffic rates, achieving high throughput and deadlock-free transaction forwarding.

The 2024 SoK also systematically reviewed the design space of payment hubs [53], analyzing the advantages of centralized hubs in transaction efficiency and relationship anonymity, as well as their limitations regarding trust assumptions and single points of failure, pointing out the trade-off relationship between scalability and security in existing hub schemes. The SoK also systematically reviewed the design space for virtual channels, covering virtual channel establishment protocols, security guarantees, and their cooperative relationship with payment channel hubs, and compared the differences among various virtual channel schemes in terms of security assumptions, script dependencies, and online requirements. Tairi et al. proposed Thunderdome, a class of rationally secure virtual channel protocols without timelocks [62]. Thunderdome uses the design philosophy of virtual channels to extend timelock-free payment channel primitives, thereby achieving multi-hop transactions without timelocks—this design not only simplifies protocol logic but also reduces the liquidity lock-up costs users face during timelock windows. Research published at the 2024 IFIP NETWORKING conference proposed an integer linear programming model to compute globally optimal VC establishment schemes across multiple dimensions such as transaction costs, security, and privacy [63]. This research paid special attention to the threat adversaries on the path pose to virtual channel security—when intermediate nodes on the VC path act maliciously, how to select the optimal VC establishment strategy becomes a key issue.

In the direction of multi-party virtual channels, Tian et al. proposed Horcrux, a general and efficient multi-party virtual channel protocol [64]. Horcrux can realize the construction of and payments through virtual channels among multiple parties without relying on additional trust assumptions, scripting languages, or permanently online requirements. The protocol adopts a novel approach called “flow neutrality," which minimizes the impact on channel balance distribution during multi-hop payments, thereby fundamentally solving the channel depletion problem.

### D. Channel Rebalancing Mechanisms

In PCNs, repeated single-direction multi-hop payments will gradually consume the balance on one side of a channel, leading to channel imbalance or even complete depletion, thereby limiting the transaction capacity and scalability of the PCN . Existing rebalancing schemes, such as sending artificial payments to rebalance channels, usually rely on on-chain operations, which reduces efficiency and limits applicable scenarios.

Kolachala et al. systematically reviewed various channel rebalancing protocols in their SoK [53], including cyclic rebalancing, hub-based rebalancing, and multi-party rebalancing, and comparatively analyzed the pros and cons of each scheme regarding on-chain operation overhead, security assumptions, and scalability.

Xu et al. proposed the Starfish rebalancing scheme, which uses the common star network structure in PCNs to achieve high-efficiency and large-capacity channel rebalancing [65]. Starfish only requires N on-chain operations to connect independent channels and aggregate the total budget of all channels. Unlike cyclic methods, Starfish does not require nodes to form a directed cycle; compared to non-cyclic methods, Starfish significantly reduces the complexity of on-chain operations and the limitations on rebalancing capacity. Dziembowski et al. proposed the Universal Channel Rebalancing (UCRb) framework, which is currently one of the most comprehensive off-chain rebalancing schemes in the literature [66]. UCRb is a blockchain-agnostic, entirely off-chain rebalancing framework that ensures correct behavior among untrusted participants without on-chain interactions.

### E. Layer-2 Rollups and Data Availability

While PCNs offer highly efficient micropayment solutions, they are largely restricted to specific applications and suffer from routing constraints [67]. In contemporary blockchain architecture, ZK-Rollups (e.g., zkSync, Starknet) and Optimistic Rollups (e.g., Arbitrum, Optimism) have largely superseded PCNs and state channels for general-purpose smart contract scalability. A comprehensive review of the current state-of-the-art must highlight Rollups.

#### 1) Optimistic Rollups

Optimistic Rollups assume by default that off-chain batch transaction execution is valid and rely on a “fraud-proof" mechanism within a predefined time window. However, centralized sequencers and Data Availability Committees (DACs) pose risks of malicious behavior, which can lead to censorship or delayed Layer-2 progression. To address this problem, Capretto, Sánchez, et al. published a study in 2025 [68] highlighting the security threats posed by dishonest sequencers and DACs to the network. They proposed and detailed a secure fraud-proof mechanism arbitrated by Layer-1 smart contracts. They concluded that through mandatory L1 arbitration and evidence generation, malicious sequencer nodes can be effectively detected and penalized, thereby fundamentally safeguarding Layer-2 users’ assets.

Furthermore, In 2024, Stefanos Chaliasos, Denis Firsov et al. [69] pointed out that introducing a Forced Queue mechanism allows users to bypass potential L2 censorship and submit transactions directly to L1, ensuring the system’s censorship resistance.

#### 2) Zero-Knowledge Rollups (ZK-Rollups) and Validiums

Unlike Optimistic Rollups, ZK-Rollups utilize advanced cryptographic validity proofs (such as ZK-SNARKs) to mathematically guarantee the correctness of off-chain computations, achieving near-instant finality. However, in addition to proof generation overhead, their limitations in privacy protection have drawn academic attention. In 2024, Torralba-Agell et al. [70] questioned the privacy flaws of ZK-Rollups. They

published a systematic analysis of existing ZK-Rollup architectures, focusing on the fact that although ZK-Rollups significantly improve throughput and reduce verification costs, their "zero-knowledge" nature often fails to provide sufficient transaction privacy in actual L2 networks. They concluded that current ZK-Rollups are designed more to achieve "succinctness" for reducing computational energy consumption rather than true user data privacy. Therefore, future L2 designs urgently need to introduce new trade-off mechanisms between throughput and privacy protection.

*3) Modular Data Availability (DA) Layers*

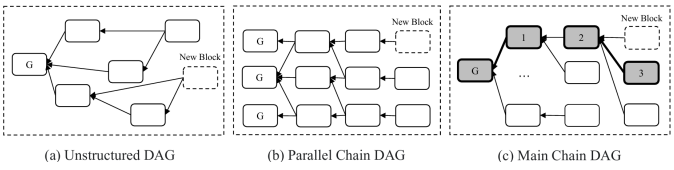
As Rollups scale, the block space on the L1 main chain for publishing transaction data becomes a severe bottleneck, driving the widespread adoption of modular architectures that decouple data availability (DA) from the execution layer. When facing massive data verification pressure, how light nodes ensure data is truly available is a core challenge. In 2024, Li Quan addressed the inefficiencies of blockchain data outsourcing and storage by publishing a new model called the "Distributed Transparent Data Layer (DTDL)." [71] This model introduced zero-knowledge proofs and Data Availability Sampling (DAS). They concluded that this approach enables nodes to store only fragments of historical records while maintaining data integrity in a malicious publisher environment, thereby achieving seamless, fork-free scaling. Table II compares various off-chain scaling technologies in terms of whether they support smart contracts, finality determination, and other aspects.

Furthermore, in 2025, Sasidharan et al. addressed the optimization problem of DAS topology efficiency in decentralized systems [72]. They published a mathematical architecture applying "Block Circulant Codes" to data availability sampling. Through rigorous derivation, they concluded that applying specific erasure code structures not only greatly reduces network communication overhead but also provides robust data recoverability guarantees for massive-scale modular blockchain networks (such as the Ethereum Danksharding roadmap).

## IV. Directed Acyclic Graph-based Blockchain Structures

To break the throughput bottleneck caused by serial transaction processing in single-chain structures, researchers introduced the Directed Acyclic Graph (DAG) structure into blockchain systems [73]. Unlike linear chains where each block references only one preceding block, the DAG structure allows a block to reference multiple preceding node blocks simultaneously, thus forming a partial order relationship globally [74].

The DAG structure significantly improves the system's parallel packing and broadcasting capabilities without destroying consistency, allowing different nodes to concurrently generate and broadcast data units within the same time window [75]. As shown in figure 2, Based on the different organizational methods and main trunk structures of nodes in a DAG, existing DAG-based blockchain structures can be roughly divided into the following three categories [76].




(a) Unstructured DAG (b) Parallel Chain DAG (c) Main Chain DAG

Fig. 2. Classification of dag-based blockchains

### A. Natural Structure DAG

A natural structure DAG does not have a fixed main branch or predefined topology, and the reference relationships between blocks are relatively free. The system usually relies on mechanisms like cumulative weight, voting strength, or probability convergence to determine the confirmation level of data and eventually form a consistent view. This type of structure has high structural flexibility, but usually requires additional mechanisms to guarantee deterministic sorting and eventual consistency.

IOTA's Tangle is a representative implementation of a natural structure DAG [77]. In Tangle, DAG nodes are defined as transactions, and peer nodes dynamically build a natural structure DAG by submitting transactions. Each new transaction must verify two existing transactions (called tips); in this way, reference relationships are formed between transactions. Tangle uses a cumulative weight mechanism for conflict resolution. Nodes calculate and propagate weights to vote among multiple candidate transaction views, eventually converging to a transaction state recognized by the entire network [78]. Tangle 2.0 further introduced a decentralized coordinator, solving the problems of the previous centralized coordinator and achieving fully decentralized consensus [79].

GraphChain is another implementation of a natural structure DAG, where each transaction needs to confirm at least two ancestor transactions and comes with a Proof of Work with optional difficulty [80]. In this way, Proof of Work accumulates and gradually confirms previous valid historical transaction records.

### B. Parallel Structure DAG

A parallel structure DAG is composed of multiple independent chains in parallel. Each chain can generate blocks independently and maintain consistency through cross-chain mechanisms or global consensus protocols [81]. This type of DAG is also the mainstream structure for implementing DAG-based BFT consensus protocols. Parallel structures usually require configuring a genesis block for each sub-chain, and new blocks generally need to be appended to the ends of the sub-chains maintained by various nodes.

HashGraph uses a protocol based on Asynchronous Byzantine Fault Tolerance (ABFT). By allowing participants to communicate in the network without relying on time synchronization, it reduces the impact of network latency and improves the system's fault tolerance [82]. In HashGraph, "events" act as nodes in the DAG, and nodes are organized via a parallel structure to ensure multiple transactions can be processed in parallel. By establishing witnesses for each event, HashGraph can establish a deterministic sequence of transactions, thereby

TABLE II
COMPREHENSIVE COMPARISON OF OFF-CHAIN AND LAYER-2 SCALING TECHNOLOGIES

| Technology | Smart Contracts | Finality | DA Location | Security Model | Use Case |
|---|---|---|---|---|---|
| Payment Channels | No | Instant | Off-chain | Multi-sig & disputes | Micro-payments |
| Optimistic Rollups | Yes (EVM) | High (7d) | On-chain (L1) | Fraud proofs | DeFi & NFTs |
| ZK Rollups | Yes (zkEVM) | Low | On-chain (L1) | Validity proofs | Enterprise finance |
| Validiums / Mod DA | Yes | Low | Off-chain | Validity proofs + DAC | Gaming & Social |

ensuring all network participants reach an agreement on the order of events.

Bullshark is a practical solution that applies the DAG structure to Byzantine Fault Tolerance protocols [83]. By combining an efficient message-passing mechanism with an optimized consensus algorithm, the protocol significantly reduces the communication complexity found in traditional BFT protocols and improves the system's fault tolerance. Bullshark improves upon the DAG-Rider protocol, achieving high performance in actual deployments [84].

LightDAG is a DAG-based, low-latency Byzantine Fault Tolerance consensus protocol aimed at solving the high-latency problem present during the broadcast process of existing DAG-based consensus algorithms [85]. LightDAG proposes two variants: LightDAG1 and LightDAG2, which optimize latency through consistent broadcasting and normal broadcasting, respectively, achieving optimal latencies of 5 steps and 4 steps.

### C. Main-chain Structure DAG

A main-chain structure DAG introduces a main chain within multiple branch structures. The main chain is responsible for global sorting or authoritative confirmation, while other branch chains can serve as a means of auxiliary storage or accelerating transaction confirmation [86]. The difference between this and parallel structures is that a main-chain structure only needs one genesis block and can be backward compatible with traditional smart contracts.

The GHOST (Greedy Heaviest Observed Subtree) protocol, originally proposed by Sompolinsky and Zohar, aims to provide an efficient Directed Acyclic Graph structure for blockchain systems [87]. The GHOST protocol uses the DAG structure to allow the blockchain to generate multiple blocks in parallel, improving throughput and solving the efficiency bottlenecks of traditional chain structures. The core idea of the GHOST protocol is that when a blockchain network forks, it selects the "heaviest" branch, or the one containing the most blocks, as the standard chain, thereby ensuring the stability of the consensus.

The Conflux system uses a DAG structure called Tree-Graph, which is an enhanced version based on the GHOST protocol [88]. Conflux adopts a recursive tree-graph structure where a pivot chain acts as a guide. The tree-graph structure merges the advantages of tree structures and graph structures: the tree structure provides simpler ordering and fewer conflicts, while the graph structure allows multiple blocks to be generated in parallel, thereby increasing the system's throughput and transaction confirmation speed. Conflux's consensus algorithm can achieve high throughput while ensuring security; tests indicate its TPS can reach over 3000 [89].

PHANTOM GHOSTDAG is a further extension of the GHOST protocol that supports the parallel processing of more blocks and performs better in high-concurrency environments [90]. By using a parameterized method, the protocol allows blocks to have multiple parent blocks, thus improving throughput while guaranteeing security.

The LBFT-DAG system uses a DAG structure called Parallel Main-Sub Chain [76], which is an enhanced version of a leader-driven Byzantine Fault Tolerance consensus protocol. LBFT-DAG proposes a streamlined DAG ledger structure consisting of a main chain and multiple parallel sub-chains. The main chain is made of blocks generated by the leader node, and each block only references two predecessor blocks, thereby reducing structural complexity. This structure combines the orderliness of the main chain with the high concurrency advantages of parallel sub-chains: the main chain provides a stable global sorting foundation, reducing conflicts, while the parallel sub-chains allow multiple nodes to generate blocks simultaneously, thereby enhancing system throughput and transaction confirmation efficiency. By asynchronously voting only on leader blocks, the consensus algorithm of LBFT-DAG achieves high throughput while ensuring security. Tests show its average transaction throughput can reach over 8000 TPS, which is about 4 times that of existing solutions.

### D. Comparative Analysis of DAG Structures

Table III provides a comparative analysis of the three types of DAG structures across multiple dimensions.

These three DAG structures focus on different aspects of topology design: Natural structure DAGs have no fixed main branch and rely on cumulative weight consensus. They have moderate implementation difficulty but relatively high confirmation delays. Parallel structure DAGs consist of multiple parallel chains, use BFT-class consensus, have the lowest confirmation delay, but the highest implementation complexity. Main-chain structure DAGs take the form of a single main chain with branches and combine with GHOST-class consensus, striking a good balance between low delay and moderate complexity.

In terms of performance, all three can achieve high throughput, but their confirmation delays and implementation difficulties vary. Natural structures (like IOTA) are highly flexible but somewhat slow to confirm; parallel structures (like Hash-Graph) have the lowest delay but the most complex engineer-

ing; main-chain structures (like Conflux) balance efficiency and feasibility through main chain anchoring.

## V. Conclusion and Future Work

### A. Conclusion

This paper systematically reviewed the on-chain scaling technologies, off-chain scaling technologies, and DAG structures of blockchains. The main research work and conclusions are as follows:

**On-chain Scaling Technologies:** Sharding technology divides the network into multiple parallel shards, achieving a linear increase in transaction throughput with the number of shards, but faces challenges regarding intra-shard consensus efficiency and load balancing. Consensus mechanism optimization improves protocol design to increase network capacity and transaction efficiency while maintaining security.

**Off-chain Scaling Technologies:** Payment channel networks move high-frequency transactions off-chain for execution, achieving millisecond-level confirmations and extremely high throughput, but require pre-depositing funds and rely heavily on network topology. Furthermore, Layer-2 Rollups (ZK and Optimistic) have superseded PCNs for general-purpose smart contracts, leveraging modular Data Availability layers to achieve massive throughput without sacrificing main-chain security.

**DAG Structures:** The DAG structure significantly improves system throughput capabilities by allowing blocks to be generated in parallel. The three DAG solutions—natural structure, parallel structure, and main-chain structure—each have their own characteristics in terms of flexibility, determinism, and implementation complexity. Natural structure DAGs (like IOTA) have high flexibility but weak determinism; parallel structure DAGs (like HashGraph) are suitable for BFT consensus; main-chain structure DAGs (like Conflux) achieve high throughput while ensuring security.

Overall, “on-chain basic optimization + off-chain and Layer-2 collaborative scaling" has become the mainstream direction in blockchain scalability research. While earlier research relied primarily on PCNs as evidence for off-chain capabilities, PCNs are largely restricted to specific applications like micropayments on the Lightning Network. The inclusion of Layer-2 Rollups provides the necessary evidence to fully support the claim that this collaborative approach is the optimal trajectory for blockchain development. On-chain optimization enhances the system’s basic processing capacity through sharding and improved consensus mechanisms; off-chain and Layer-2 architectures achieve nearly infinite throughput by moving transactions off the main chain. The two types of solutions complement each other to build a multi-layered blockchain scaling system.

### B. Future Outlook

Although significant progress has been made in blockchain scalability research, many challenges remain. Future explorations can proceed in the following directions:

**Multi-technology Fusion and Collaborative Optimization:** Existing scaling technologies each have their pros and cons. Future research could explore integrating DAG structures with sharding technology, using DAG’s parallelism to enhance processing capacity within shards while coordinating across DAG subgraphs through a sharding architecture [91]. Consensus mechanism optimization could also collaborate with payment channel networks [92] to improve off-chain transaction efficiency while ensuring security.

**Adaptive and Intelligent Scaling Mechanisms:** Introducing Artificial Intelligence and Machine Learning technologies to achieve adaptive shard adjustments, dynamic consensus parameter configurations, and intelligent routing selection [93]. Rebalancing strategies based on Reinforcement Learning could dynamically optimize fund scheduling according to real-time transaction flow patterns, significantly improving channel utilization and transaction success rates.

**Privacy Protection Enhancement:** Combining cryptographic technologies like Zero-Knowledge Proofs and Homomorphic Encryption to design scaling solutions with enhanced privacy protection. For example, the Shaduf++ scheme introduces commitment schemes and range proofs to safely store bonded amounts encrypted, protecting user privacy [94].

**Standardization of Cross-chain Interoperability:** Promoting the standardization of cross-chain protocols and designing universal cross-chain communication frameworks to achieve seamless interoperability between heterogeneous blockchains. Cross-chain interoperability standards should include core elements such as message formats, verification mechanisms, security assumptions, and incentive models [95].

**Formal Verification and Security Proofs:** Utilizing formal verification tools to perform machine-verifiable proofs on scaling protocols to ensure their security under various attack models. For complex systems that combine multiple scaling technologies, establishing a composable security framework is necessary [96].

**Engineering Deployment and Practical Verification:** Deploying and verifying proposed schemes in real blockchain networks (like the Bitcoin Lightning Network and Ethereum), and further optimizing algorithm parameters using real-world environment data to enhance the engineering practical value of the schemes [97].

## Acknowledgment

This work was supported by the Key Research and Development Program of Shandong Province of China (Grant No.2023CXPT056).

## References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4(2):15, 2008.

[2] Thippa Reddy Gadekallu, Thien Huynh-The, Weizheng Wang, Gokul Yenduri, Pasika Ranaweera, Quoc-Viet Pham, Daniel Benevides da Costa, and Madhusanka Liyanage. Blockchain for the metaverse: A review. *arXiv preprint arXiv:2203.09738*, 2022.

[3] Maher Boughdiri, Takoua Abdellatif, and Chirine Ghedira Guegan. A systematic literature review on blockchain storage scalability. *IEEE Access*, 2025.

[4] Wei Tong, Jian Li, Lingxiao Yang, Xiyan Huang, Xiangshang Gao, and Zesong Dong. A survey on blockchain scalability. In *2024 International Conference on Blockchain, Metaverse and Trustworthy Systems (BlockSys)*, pages 133–147. Springer Nature Singapore, 2024.

TABLE III
COMPARISON OF DAG STRUCTURE CHARACTERISTICS

| Characteristic | Natural Structure DAG | Parallel Structure DAG | Main-chain Structure DAG |
|---|---|---|---|
| Topology | No fixed main branch | Multiple parallel chains | Single main chain with forks |
| Deterministic Sorting | weak | Strong | Strong |
| Consensus Mechanism | Cumulative Weight | BFT Class Consensus | GHOST Class Consensus |
| Representative Systems | IOTA, GraphChain | HashGraph, Bullshark | Conflux, GHOST |
| Throughput | High | High | High |
| Confirmation Delay | Moderate | Low | Low |
| Implementation Complexity | Moderate | High | Moderate |


[5] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains: (a position paper). In *International conference on financial cryptography and data security*, pages 106–125. Springer, 2016.

[6] Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. Sok: Dag-based blockchain systems. *ACM Computing Surveys*, 55(12):1–38, 2023.

[7] Diana Hawashin, Mohamed Nemer, Senay A Gebreab, Khaled Salah, Raja Jayaraman, Muhammad Khurram Khan, and Ernesto Damiani. Blockchain applications in uav industry: Review, opportunities, and challenges. *Journal of Network and Computer Applications*, 230:103932, 2024.

[8] Yuqi Xie, Changbing Tang, Zhiyi Chen, Jingang Lai, Zhonglong Zheng, and Xinghuo Yu. Towards proof-of-prospect consensus mechanism for maximizing consumers' satisfaction in distributed energy systems. *Science China Information Sciences*, 69(2):122202, 2026.

[9] Ziqiang Xu, Ahmad Salehi Shahraki, and Naveen Chilamkurti. A fault-tolerant sharding mechanism for resilience and scalability in blockchain using backup pool. *Future Generation Computer Systems*, 174:107982, 2026.

[10] Asmaa Aldoubaee, Noor Hafizah Hassan, and Fiza Abdul Rahim. A systematic review on blockchain scalability. *International Journal of Advanced Computer Science and Applications*, 14(9):774–784, 2023.

[11] Jianting Zhang, Zicong Hong, Xiaoyu Qiu, Wuhui Chen, Yufeng Zhan, and Song Guo. Scaling blockchain via dynamic sharding. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[12] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. Sok: Sharding on blockchain. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 41–61, 2019.

[13] Brandon Liew Yi Quan, Nur Haliza Abdul Wahab, Arafat Al-Dhaqm, Ahmad Alshammari, Ali Aqarni, Shukor Abd Razak, and Koh Tieng Wei. Recent advances in sharding techniques for scalable blockchain networks: A review. *IEEE Access*, 13:21335–21366, 2024.

[14] Fahad Rahman, Chafiq Titouna, and Farid Naït-Abdesselam. Efficient shard load balancing for scalable blockchain systems. *Cluster Computing*, 28(8):506, 2025.

[15] Yizhong Liu, Andi Liu, Zhuocheng Pan, Yuxuan Hu, Jianwei Liu, Song Bian, Yuan Lu, Zhenyu Guan, Dawei Li, and Meikang Qiu. Realizing corrupted-shard tolerance: A sharding blockchain with preserving global resilience. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2099–2113, 2025.

[16] Chang Liu, Kang Ning, P Takis Mathiopoulos, Zheng Xue, and Guojun Han. A novel dual-layer multi-shard blockchain architecture for vehicle data sharing. *IEEE Transactions on Mobile Computing*, 2026.

[17] Li Lu, Linfu Sun, and Yisheng Zou. An efficient sharding consensus protocol for improving blockchain scalability. *Computer Communications*, 231:108032, 2025.

[18] Yan Wang, Zheng Gong, Dayu Jia, Aiping Tan, and Minchao Liu. Dynamic sharding model and performance optimization method for consortium blockchain: Y. wang et al. *The Journal of Supercomputing*, 81(2):411, 2025.

[19] Luyi Zhang, Yujue Wang, Yong Ding, Hai Liang, Changsong Yang, and Chunhai Li. Sharding technologies in blockchain: Basics, state of the art, and challenges. In *International Conference on Blockchain and Trustworthy Systems*, pages 242–255. Springer, 2023.

[20] Huawei Huang, Zhaokang Yin, Qinde Chen, Jian Zheng, Xiaofei Luo, Guang Ye, Xiaowen Peng, Zibin Zheng, and Song Guo. Brokerchain: A blockchain sharding protocol by exploiting broker accounts. *IEEE Transactions on Networking*, 2025.

[21] Xiaorong Zhu, Zhiwei Yuan, and Qinyin Ni. A self evolving high performance sharded consortium blockchain designed for secure and trusted resource sharing for 6g networks. *Scientific Reports*, 15(1):28098, 2025.

[22] Huawei Huang, Jian Zheng, Xuanye Zhu, Yue Lin, Ting Cai, Lu Zhou, Zibin Zheng, and Song Guo. Shardcutter: A blockchain sharding protocol achieving transaction workload balance across state shards. *IEEE Transactions on Networking*, 2026.

[23] Hao Xu, Jiaqi Zhang, Xiulong Liu, Zhimin Yu, Tingyu Fan, Baochao Chen, and Keqiu Li. Tangram: Enabling efficient and balanced dynamic storage extension on sharding blockchain systems. *IEEE Transactions on Computers*, 2025.

[24] Guoqiong Liao, Yinxiang Lei, Yufang Xie, and Neal N Xiong. Lsbsp: A lightweight sharding method of blockchain based on state pruning for efficient data sharing in iomt. *Computers, Materials & Continua*, 82(2), 2025.

[25] Kaimin Zhang, Xingwei Wang, Bo Yi, Min Huang, Lin Qiu, Enliang Lv, and Jingjing Guo. A reliable distributed-cloud storage based on permissioned blockchain. *IEEE Transactions on Services Computing*, 18(3):1216–1231, 2025.

[26] Nana Akua Nkansa Adu-Amankwa and Farzad Rahimian. Harnessing blockchain-enabled digital twins for building commissioning: Examining practitioners' perspectives. *Engineering, Construction and Architectural Management*, 33(3):1891–1906, 2026.

[27] A Simbu, S Nandakumar, and K Saravanan. Blockchain-driven smart contract with key exchange protocol for secure device-to-device communication using verkle tree k-ary structures. *Scientific Reports*, 2026.

[28] Jan Oberst. Towards stateless clients in ethereum: Benchmarking verkle trees and binary merkle trees with snarks. *arXiv preprint arXiv:2504.14069*, 2025.

[29] Lei Tian, Feilong Lin, Jiahao Gan, Jiahao Qi, Riheng Jia, Zhonglong Zheng, Minglu Li, and Xuemin Shen. Enhancing security in parallel federated learning with sharded blockchain for internet of vehicles. *IEEE Transactions on Vehicular Technology*, 2025.

[30] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 931–948, 2018.

[31] Xiangdong Guo, Yuting Zhang, Jingfa Yao, Guifa Teng, et al. Performance optimization of agricultural traceability blockchain based on sharding technology. *Quality Assurance and Safety of Crops & Foods*, 17(2):213–231, 2025.

[32] Yan Wang, Hao Wang, and Yanghuang Cao. Comprehensive review of storage optimization techniques in blockchain systems. *Applied Sciences*, 15(1):243, 2024.

[33] Xiaofei Luo, Junhao Wu, Baozhou Xie, Jian Zheng, Qinde Chen, and Huawei Huang. A survey of sharding blockchains on performance and scalability. pages 105–118, 2025.

[34] Mahran Morsidi, Sharul Tajuddin, SH Shah Newaz, Ravi Kumar Patchmuthu, and Gyu Myoung Lee. A review on blockchain sharding for improving scalability. *Future Internet*, 17(10):481, 2025.

[35] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE communications surveys & tutorials*, 22(2):1432–1465, 2020.

[36] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consen-

sus in the age of blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 183–198, 2019.

[37] Xunqiang Lan, Xiao Tang, Ruonan Zhang, Bin Li, Qinghe Du, Dusit Niyato, and Zhu Han. Distributionally robust game for proof-of-work blockchain mining under resource uncertainties. *IEEE Transactions on Information Forensics and Security*, 21:1036–1049, 2026.

[38] Zhijun Wu, Hanwen Xu, Zhiquan Liu, Meng Yue, and Yanrong Lu. Defending pow blockchains against game-theoretic dos attacks: A rational strategy analysis. *IEEE Transactions on Dependable and Secure Computing*, 2026.

[39] Bolun Yang, Jiamin Hao, Yao Ma, and Li Zhou. The optimal mining strategy of proof of stake consensus in peercoin blockchain. *Electronics*, 15(5):974, 2026.

[40] Shymaa M Arafat. A study of blockchain consensus protocols. *Cybersecurity*, 9(1):120, 2026.

[41] Kevin Werbach. Summary of blockchain governance in the wild. In *The Blockchain Scholars Book: Current Academic Insights Condensed for Busy Practitioners*, pages 143–149. Springer, 2026.

[42] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186, 1999.

[43] Dakai Kang, Suyash Gupta, Dahlia Malkhi, and Mohammad Sadoghi. Hotstuff-1: Linear consensus with one-phase speculation. *Proceedings of the ACM on Management of Data*, 3(3):1–29, 2025.

[44] Ali Ahmed Al-awamy, Nagi Al-shaibany, Axel Sikora, and Dominik Welte. Hybrid consensus mechanisms in blockchain: a comprehensive review. *International Journal of Intelligent Systems*, 2025(1):5821997, 2025.

[45] Jianhua Liu, Xin Wang, Guangtao Xue, Tong Liu, and Minglu Li. An eca regret learning game for cross-tier computation offloading against swarm attacks in sensor edge cloud. *IEEE Internet of Things Journal*, 11(1):1201–1216, 2023.

[46] Soumen Giri, Manasi Roy, Sujoy Bandyopadhyay, Sourav Das, Sutanu Mangal, and Subhankar Samanta. Metal–organic frameworks: classifications, synthesis, structure–property–performance relationship, and techno-economic analysis of redox flow batteries. *Journal of Materials Chemistry A*, 13(25):19193–19235, 2025.

[47] Yuhui Huo, Xiaolu Luo, Zhen Lu, Yihan Lu, Qi Cai, Miao Hu, and Dong Tang. Scaling efficiency and incentive mechanism optimization in the bitcoin lightning network. In *2025 lEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 1–8. IEEE, 2025.

[48] Nichole King-Price. Optimizing global micropayments through bitcoin's on-chain architecture for enhanced scalability and economic efficiency. 2026.

[49] Stefan Dziembowski, Sebastian Faust, and Kristina Hostáková. General state channel networks. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 949–966, 2018.

[50] Vibhaalakshmi Sivaraman, Shaileshh Bojja Venkatakrishnan, Kathleen Ruan, Parimarjan Negi, Lei Yang, Radhika Mittal, Giulia Fanti, and Mohammad Alizadeh. High throughput cryptocurrency routing in payment channel networks. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 777–796, 2020.

[51] Lingxiao Yang, Xuewen Dong, Wei Tong, Shiyang Ma, Hui Qiao, and Siyu Xing. Secure off-chain payment in consortium blockchain system. In *2020 International Conference on Networking and Network Applications (NaNA)*, pages 259–264. IEEE, 2020.

[52] Rene Pickhardt. A mathematical theory of payment channel networks. *arXiv preprint arXiv:2601.04835*, 2026.

[53] Kartick Kolachala, Mohammed Ababneh, and Roopa Vishwanathan. Sok: Payment channel networks. In *2024 Conference on Building a Secure & Empowered Cyberspace (BuildSEC)*, pages 28–35. IEEE, 2024.

[54] David Lobmaier, Rafael Konlechner, Stefan Schulte, and Ingo Weber. Assessing routing algorithms for payment channel networks. *Distributed Ledger Technologies: Research and Practice*, 3(1):1–27, 2024.

[55] Qiushi Wei, Yuhui Zhang, Dejun Yang, and Guoliang Xue. Bar: A balance-aware routing protocol in payment channel networks. In *ICC 2025-IEEE International Conference on Communications*, pages 2647–2652. IEEE, 2025.

[56] Longxia Huang, Changzhi Huo, Chengzhi Ge, and Mengmeng Yang. Aero: An adaptive and efficient routing for off-chain payment channel networks. *Computer Networks*, 257:111009, 2025.

[57] Padraig Corcoran and Irena Spasić. Path planning at the edge of payment channel networks. *Distributed Ledger Technologies: Research and Practice*, 2025.

[58] Hasan Falah Hasan, Mohd Najwadi Yusoff, Je Sen Teh, and Shams Mhmood Abdali. Emcpf: Efficient multi-party conditional payment flow payment protocol on lightning payment channel network. *Blockchain: Research and Applications*, page 100359, 2025.

[59] Xiaoxue Zhang and Chen Qian. A flexible cross-chain payment channel network. *IEEE Transactions on Networking*, 2026.

[60] Lingxiao Yang, Xuewen Dong, Wei Wang, Yong Yu, Sheng Gao, Qiang Qu, and Yulong Shen. Share: Optimizing secure hub allocation and routing efficiency in payment channel networks. *arXiv preprint arXiv:2501.04236*, 2025.

[61] Lingxiao Yang, Xuewen Dong, Sheng Gao, Qiang Qu, Xiaodong Zhang, Wensheng Tian, and Yulong Shen. Optimal hub placement and deadlock-free routing for payment channel network scalability. In *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, pages 692–702. IEEE, 2023.

[62] Zeta Avarikioti, Yuheng Wang, and Yuyi Wang. Thunderdome:{Timelock-Free}{Rationally-Secure} virtual channels. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 4187–4204, 2025.

[63] Lukas Aumayr, Esra Ceylan, Yannik Kopyciok, Matteo Maffei, Pedro Moreno-Sanchez, Iosif Salem, and Stefan Schmid. Optimizing virtual payment channel establishment in the face of on-path adversaries. *Computer Communications*, 238:108155, 2025.

[64] Anqi Tian, Peifang Ni, Yingzi Gao, and Jing Xu. Horcrux: Synthesize, split, shift and stay alive; preventing channel depletion via universal and enhanced multi-hop payments. *IACR Cryptol. ePrint Arch.*, 2024:1338, 2024.

[65] Minghui Xu, Wenxuan Yu, Guangyong Shang, Guangpeng Qi, Dongliang Duan, Shan Wang, Kun Li, Yue Zhang, and Xiuzhen Cheng. Starfish: Rebalancing multi-party off-chain payment channels. *arXiv preprint arXiv:2504.20536*, 2025.

[66] Stefan Dziembowski, Shahriar Ebrahimi, Omkar Gavhane, and Susil Kumar Mohanty. Universal channel rebalancing: Flexible coin shifting in payment channel networks. *Cryptology ePrint Archive*, 2025.

[67] Yevhenii Shcherbina. Comparative analysis of layer 2 scaling techniques: Rollups, sidechains and lightning network in blockchain systems. *Emerging Frontiers Library for The American Journal of Engineering and Technology*, 7(10):75–82, 2025.

[68] Margarita Capretto, Martín Ceresa, Antonio Fernandez Anta, Pedro Moreno-Sanchez, and César Sánchez. A secure sequencer and data availability committee for rollups. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2129–2143, 2025.

[69] Stefanos Chaliasos, Denis Firsov, and Benjamin Livshits. Towards a formal foundation for blockchain zk rollups. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2714–2728, 2025.

[70] Adrià Torralba-Agell, Ghazaleh Keshavarzkalhori, Cristina Pérez Solà, David Megías, Jordi Herrera Joancomartí, et al. Unmasking the illusion: The shortcomings of "zero-knowledge" rollups in achieving privacy. *XVIII Reunión Española sobre Criptología y Seguridad de la Información: XVIII RECSI*, pages 361–366, 2024.

[71] Li Quan. Distributed transparent data layer for next generation blockchains. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1178–1181, 2024.

[72] Birenjith Sasidharan, Emanuele Viterbo, and Son Hoang Dau. Block circulant codes with application to decentralized systems. *IEEE Transactions on Communications*, 2025.

[73] Antonio Villafranca, Igor Tasic, Victor Gallegos, Almudena Gimenez, Jesus Ochoa Rego, Juan Antonio Fernandez, and Maria-Dolores Cano. Comparing blockchain and dag technologies for smart agriculture traceability in terms of efficiency and latency. *Simulation Modelling Practice and Theory*, 142:103131, 2025.

[74] Anderson Melo de Morais, Fernando Antonio Aires Lins, and Nelson Souto Rosa. Selecting consensus algorithm integrations in a dag-based blockchain for iot using genetic algorithms. *Journal of Internet Services and Applications*, 17(1):55–71, 2026.

[75] Lang Li, Dongyan Huang, and Chengyao Zhang. An efficient dag blockchain architecture for iot. *IEEE Internet of Things Journal*, 10(2):1286–1296, 2022.

[76] Xuewen Dong, Yi Liu, Teng Li, Xiaojie Guo, Youliang Tian, Yulong Shen, and Xiaojiang Du. Lbft-dag: A swift, leader-driven, dag-based consortium blockchain with byzantine fault-tolerance. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2025.

[77] Wellington Fernandes Silvano and Roderval Marcelino. Iota tangle: A cryptocurrency to communicate internet-of-things data. *Future generation computer systems*, 112:307–319, 2020.

[78] Sebastian Müller, Andreas Penzkofer, Nikita Polyanskii, Jonas Theis, William Sanders, and Hans Moog. Tangle 2.0 leaderless nakamoto consensus on the heaviest dag. *IEEE Access*, 10:105807–105842, 2022.

[79] Peilin He, Tao Yan, Chuanshan Huang, Nicolò Vallarano, and Claudio J Tessone. Welcome to the tangle: An empirical analysis of iota ledger in the chrysalis stage. In *International Congress on Blockchain and Applications*, pages 419–431. Springer, 2023.

[80] Vishnu Kumar. Interoperable knowledge graphs for localized supply chains: Leveraging graph databases and rdf standards. *Logistics*, 9(4):144, 2025.

[81] Hao Yin, Changling Zhou, Weiping Wen, and Yiwei Liu. Fountain: Dag-based separate bft consensus made hashgraph practical. *IEEE Transactions on Network and Service Management*, 2025.

[82] Wanli Zhao and Satoshi Fujita. Hashgraph over the edge: Achieving byzantine fault tolerance in decentralized p2p frameworks. *IEEE Access*, 2025.

[83] Alexander Spiegelman, Neil Giridharan, Alberto Sonnino, and Lefteris Kokoris-Kogias. Bullshark: Dag bft protocols made practical. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2705–2718, 2022.

[84] Idit Keidar, Eleftherios Kokoris-Kogias, Oded Naor, and Alexander Spiegelman. All you need is dag. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 165–175, 2021.

[85] Xiaohai Dai, Guanxiong Wang, Jiang Xiao, Zhengxuan Guo, Rui Hao, Xia Xie, and Hai Jin. Lightdag: A low-latency dag-based bft consensus through lightweight broadcast. In *2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 998–1008. IEEE, 2024.

[86] Vivek Bagaria, Sreeram Kannan, David Tse, Giulia Fanti, and Pramod Viswanath. Prism: Deconstructing the blockchain to approach physical limits. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 585–602, 2019.

[87] Hao Yang, Jing Chen, Kun He, Meng Jia, Kai Li, Chen Zhang, and Ruiying Du. Power: High-throughput blockchain based on computing power utilization. *IEEE Transactions on Networking*, 2026.

[88] Chenxin Li, Peilun Li, Dong Zhou, Zhe Yang, Ming Wu, Guang Yang, Wei Xu, Fan Long, and Andrew Chi-Chih Yao. A decentralized blockchain with high throughput and fast confirmation. In *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, pages 515–528, 2020.

[89] Cheng Zhang, Yang Xu, Xiaowei Wu, En Wang, Hongbo Jiang, and Yaoxue Zhang. A semi-asynchronous decentralized federated learning framework via tree-graph blockchain. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 1121–1130. IEEE, 2024.

[90] Yonatan Sompolinsky, Shai Wyborski, and Aviv Zohar. Phantom ghostdag: a scalable generalization of nakamoto consensus: September 2, 2021. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pages 57–70, 2021.

[91] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. Caper: a cross-application permissioned blockchain. *Proceedings of the VLDB Endowment*, 12(11), 2019.

[92] Zesong Dong, Wei Tong, Ailiang Zhong, Xiaojie Guo, Lingtao Xue, and Xuewen Dong. A secure and robust blockchain-based identity management scheme oriented to uav network. *Blockchain: Research and Applications*, page 100439, 2026.

[93] Shruti Mishra, Mohit Bhuria, Hitesh Singla, Sujata Pal, and Isaac Woungang. Relearn: Reinforcement learning enhanced profitable rebalancing in payment channel networks. In *2025 21th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–6. IEEE, 2025.

[94] Zhonghui Ge, Chenke Wang, Yu Long, and Dawu Gu. Shaduf++: Non-cycle and privacy-preserving payment channel rebalancing. *IEEE Transactions on Dependable and Secure Computing*, 22(2):1281–1298, 2024.

[95] Debasis Mohanty, Divya Anand, Hani Moaiteq Aljahdali, and Santos Gracia Villar. Blockchain interoperability: Towards a sustainable payment system. *Sustainability*, 14(2):913, 2022.

[96] Hu Xiong, Yan Wu, Chuanjie Jin, and Saru Kumari. Efficient and privacy-preserving authentication protocol for heterogeneous systems in iiot. *IEEE Internet of Things Journal*, 7(12):11713–11724, 2020.

[97] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. Untangling blockchain: A data processing view of blockchain systems. *IEEE transactions on knowledge and data engineering*, 30(7):1366–1385, 2018.