


Binary Tree Blockchain of Decomposed Transactions

Davut Çulha

(Atılım University, Ankara, Türkiye)

 <https://orcid.org/0000-0001-5486-1867>, davut.culha@atilim.edu.tr

Abstract: Widespread adoption of blockchain technologies requires scalability. To achieve scalability, various methods are applied, including new consensus algorithms, directed acyclic graph solutions, sharding solutions, and off-chain solutions. Sharding solutions are particularly promising as they distribute workload across different parts of the blockchain network. Similarly, directed acyclic graphs use graph data structures to distribute workload effectively. In this work, a binary tree data structure is used to enhance blockchain scalability. Binary trees offer several advantages, such as the ability to address nodes with binary numbers, providing a straightforward and efficient method for identifying and locating nodes. Each node in the tree contains a block of transactions, which allows for transactions to be directed to specific paths within the tree. This directionality not only increases scalability by enabling parallel processing of transactions but also ensures that the blockchain can handle a higher volume of transactions without becoming congested. Moreover, transactions are decomposed into transaction elements, improving the immutability of the binary tree blockchain. This novel decomposition process helps to minimize the computational overhead required for calculating account balances, making the system more efficient. By breaking down transactions into their fundamental components, the system can process and verify transactions more rapidly and accurately. This approach effectively realizes implicit sharding using a binary tree structure, distributing the processing load more evenly and reducing bottlenecks. The proposed method is simulated to assess its performance. Experimental results demonstrate that the proposed method achieves a significantly higher transaction throughput of 32307 transactions per second. Furthermore, block generation times decrease as the system scales, with an average block generation time of 0.13 seconds, underscoring the efficiency of the binary tree blockchain structure.

Keywords: blockchain, binary tree blockchain, blockchain path, transaction element, UTXO, DAG, blockchain scalability, sharding, transaction element decomposition

Categories: J.0, C.2.4, K.4.4, D.4.6, D.4.8

DOI: 10.3897/jucs.135666

1 Introduction

Blockchain technologies have three important properties: decentralization, security, and scalability. However, these three properties cannot be simultaneously optimized, a concept known as the blockchain trilemma [Sanka, 21]. Despite their potential, the widespread adoption of blockchain technologies has been largely impeded by scalability issues. These technologies, also referred to as Distributed Ledger Technologies (DLTs), are assessed based on metrics such as throughput and latency [Kahmann, 23]. Modern DLTs fall into three main categories: blockchain, Directed Acyclic Graph (DAG), and hybrid DLTs. While blockchain and hybrid DLTs perform well in small network settings, DAG-based DLTs excel in environments requiring

dynamic scaling, enhancing throughput while keeping latency low in large network scenarios.

In this work, a binary tree, a specialized variant of DAG, is utilized to achieve scalability. This data structure provides several advantages, such as the ability to address nodes using binary numbers, offering a simple and efficient mechanism for identifying and locating nodes. Each node in the tree corresponds to a block of transactions, enabling the routing of transactions to specific paths within the tree. This hierarchical organization inherently enhances scalability. Unlike general DAG-based blockchains, which often require complex processes to handle transaction dependencies and ordering, the binary tree's deterministic structure simplifies these operations.

Furthermore, the binary tree refines traditional DAG approaches by breaking transactions into smaller components called transaction elements. This decomposition improves immutability and reduces the complexity of balance calculations. By ensuring that all related transaction data is stored and processed within a specific path, the binary tree minimizes cross-path conflicts and enhances data consistency. Compared to other DAG-based frameworks, such as IOTA or Fantom, the binary tree offers a more straightforward implementation, delivering comparable scalability benefits while avoiding the significant computational overhead associated with graph traversal and conflict resolution [Kahmann, 23] [Li, 21]. Additionally, the binary tree naturally facilitates implicit sharding by evenly distributing workloads across its branches, reducing bottlenecks and enabling parallel transaction processing. This seamless integration of simplicity, scalability, and efficiency positions the binary tree blockchain as a powerful and practical alternative to existing DAG-based systems.

The analysis of popular blockchain systems involves both macro and micro-level examinations [Xu, 18]. Macro-level analysis focuses on improving overall blockchain performance, while micro-level analysis dissects blockchain systems into fundamental components: network, consensus, and distributed ledger. The approach provides a modular framework for studying blockchain systems, encouraging flexible, scalable, and adaptive designs. The framework can be extended to OSI-like layers such as platform, network, consensus, data, and application layers, to analyze blockchain technologies in a layered model.

The design and performance of data structures in DLTs are critical, with structures classified as either chain-based or DAG-based. DAG-based structures are particularly effective for scalability and high throughput. These structures allow for more parallel processing of transactions, which significantly enhances the overall efficiency of the network. Additionally, security issues and potential attacks on various DLT structures are crucial. It is essential to thoroughly understand and address vulnerabilities in both chain-based and DAG-based structures to ensure the integrity and reliability of DLT systems.

Consensus algorithms play a vital role in enabling decentralized decision-making processes in blockchain systems, especially important for IoT applications with resource constraints [Singh, 22]. Given the challenges with scalability in blockchain technologies, this importance has led to the development of blockchain simulators for testing and evaluation. While existing simulators are primarily designed for Proof-of-Work (PoW) systems, there is a need for more comprehensive simulators for other blockchain types to accurately evaluate their performance.

Scalability remains a major concern in blockchain technology, impacting throughput, latency, and storage. Various solutions are being explored, including both

on-chain and off-chain solutions, aiming to improve core blockchain elements, transfer workload outside the blockchain, and enhance throughput through optimized consensus and data structures. Sharding shows promise by dividing network power into independent parts (shards) to address scalability challenges. Throughput optimization can also be achieved using layer-2 solutions, which handle transactions off the main chain (layer-1). In this work, sharding is realized implicitly by evenly distributing workload over a binary tree.

Sharding technology, originally used in databases, has been adapted to blockchain to increase transaction throughput and solve the blockchain impossible triangle problem [Liu, 22]. Various sharding blockchains and their design models are analyzed, evaluating their reliability, applicability, and scalability.

Research on blockchain scalability involves a systematic investigation of core blockchain elements, applications, and existing scalability solutions. While Byzantine Fault Tolerance (BFT) consensus is recognized for its performance and energy efficiency, scaling BFT consensus to large environments presents challenges. Efforts to enhance blockchain scalability through innovative techniques and protocols are evaluated, focusing on the trade-offs and assumptions involved.

Transaction models in blockchain can be categorized into Unspent Transaction Output (UTXO) and account models, each with its advantages for scalability and transaction processing [He, 21]. Solutions to blockchain scalability issues include on-chain and off-chain strategies, sharding, consensus mechanisms, and DAG-based solutions. Novel approaches addressing cross-shard transactions offer promising advancements in blockchain scalability. In this work, UTXO model is used to distribute transactions over a binary tree for scalability.

Continuous development and exploration of advanced solutions are crucial for addressing the scalability challenges that hinder the widespread adoption of blockchain technologies.

The rest of the paper is structured as follows: related work is presented before the proposed system model. Experimental results of the model are then provided and discussed. The paper concludes with a final section.

2 Related Work

Blockchain technologies are characterized by three key properties: decentralization, security, and scalability. However, achieving improvements in all these areas simultaneously is challenging due to the blockchain trilemma [Sanka, 21]. Blockchain technologies have not been adopted widely mainly because of scalability problems [Sanka, 21].

Distributed Ledger Technologies (DLTs) are evaluated in terms of throughput and latency [Kahmann, 23]. Modern DLTs are classified into blockchain, DAG, and hybrid models, with DAG-based systems excelling in dynamically scaling networks. IOTA [Popov, 18] is recognized as a modern DLT that improves throughput while maintaining low latency in large-scale networks.

Blockchain systems are analyzed at macro and micro levels [Xu, 18]. The macro-level focuses on enhancing performance, while the micro-level dissects blockchain into its network, consensus, and ledger components. A modular framework is introduced to facilitate scalable and adaptable blockchain system design.

The structure of DLT data models is also examined [Wu, 22], categorizing them into chain-based and DAG-based architectures. A detailed taxonomy of these structures is proposed. Hashgraph [Baird, 16], a parallel chain-based cryptocurrency, employs the gossip protocol for consensus. DAG-based architectures are widely adopted for higher scalability and throughput, though security concerns and potential attacks are also explored. Similarly, the proposed binary tree blockchain structure enhances scalability by leveraging hierarchical transaction distribution, allowing efficient workload balancing and parallel transaction processing across the network.

Consensus algorithms are essential for decentralized decision-making, especially in IoT applications with limited resources [Singh, 22]. Various mechanisms, including Proof-of-X, RAFT [Ongaro, 14], BFT, Paxos [Lamport, 01], and DAG-based algorithms, are analyzed in terms of their benefits, limitations, and suitability for IoT environments.

Blockchain technologies encounter performance limitations that hinder their broader adoption. Blockchain simulators provide a cost-effective means of assessing performance by emulating real-world scenarios. However, most existing simulators are designed for PoW-based systems and remain underdeveloped [Paulavičius, 21]. Many focus on isolated aspects of blockchain functionality, and no single simulator comprehensively represents all blockchain characteristics [Paulavičius, 2021], underscoring the need for more advanced and inclusive tools.

Scalability remains a fundamental challenge in blockchain technology, with throughput, latency, and storage being key concerns [Sanka, 21]. Research has explored scalability issues, including energy consumption and cost inefficiencies [Khan, 21]. In industrial and IoT applications, throughput and latency are particularly critical. Scalability solutions fall into five categories: on-chain, off-chain, consensus mechanisms, DAG-based, and sharding [Matani, 24]. On-chain solutions, such as SegWit, sharding, and improved consensus models, enhance core blockchain components, while off-chain solutions like the Lightning Network offload processing from the blockchain. DAG-based approaches restructure data organization, whereas consensus mechanisms focus on increasing throughput. Among these, sharding, which partitions the network into independent processing units, is regarded as the most promising approach for scalability enhancement [Matani, 24] [Nasir, 22].

In this work, sharding is achieved by evenly distributing workloads through a binary tree structure, enabling efficient node addressing with binary numbers. Each node corresponds to a block of transactions, facilitating transaction routing along predefined paths to enhance scalability. Transactions are decomposed into smaller elements, strengthening immutability and streamlining balance calculations. This method implements implicit sharding by evenly distributing processing loads, thereby mitigating bottlenecks.

Blockchain scalability is analyzed across three dimensions: improvements to the core blockchain, scalability solutions within applications, and a broader examination of blockchain solutions [Nasir, 22]. BFT consensus is recognized for its high performance, energy efficiency, and correctness, making it a strong candidate for scalable blockchain systems. However, its application in large-scale environments presents challenges, prompting research into more advanced BFT protocols [Berger, 23].

Sharding mechanisms in blockchain are systematically examined, emphasizing key components such as node selection, intra-shard consensus, and cross-shard transaction handling [Liu, 22]. Existing sharding-based systems, including Omniledger [Kokoris-

Kogias, 18], ELASTICO [Luu, 16], Monoxide [Wang, 19], and RapidChain [Zamani, 18], are evaluated for their impact on transaction throughput and scalability. Different types of sharding, such as communication, computation, and storage sharding, are also explored. In contrast to these explicit sharding approaches, the proposed method achieves implicit sharding by leveraging a binary tree structure, where transactions are automatically distributed across hierarchical paths based on their associated account addresses, reducing cross-shard communication overhead.

Sharding models are categorized based on blockchain type and methodology [Li, 23]. A comparative analysis of various sharding schemes is conducted, identifying security vulnerabilities and recommending countermeasures. Additionally, evaluation criteria for reliability, applicability, and scalability are established, offering a structured approach to assessing sharding implementations [Liu, 23].

Recent developments in sharding technologies, including Polkadot [Abbas, 22], Ethereum Casper [Bachani, 22], and Cardano Hydra [Jourenko, 22], are analyzed, emphasizing blockchain performance trade-offs and challenges [Monrat, 23]. Polkadot utilizes a sharded blockchain structure with "parachains" linked to a central "Relay Chain," enabling parallel processing and interoperability. Ethereum's scalability limitations arise from requiring nodes to maintain the entire blockchain, whereas Ethereum Casper enhances scalability and security through a Beacon Chain that coordinates shard chains. Cardano Hydra, a layer-2 scaling solution, improves Cardano's throughput without necessitating inter-shard synchronization. Key challenges in sharding include shard locality planning, discovery mechanisms, and efficient inter-shard routing.

Blockchain transaction models are primarily divided into UTXO and account-based approaches [He, 21] [Matani, 24] [Müller, 23]. Bitcoin's UTXO model [Nakamoto, 08] processes transactions atomically, facilitating parallel execution and improved scalability. Conversely, Ethereum's account model [Buterin, 13] maintains balances in a global state, streamlining transaction validation. EZchain introduces a hybrid value model [Xue, 23], combining features of both models by managing outputs similarly to the account model while eliminating the need for separate unspent outputs. In this work, the UTXO model is employed within a binary tree structure to enhance scalability.

Scalability solutions encompass both on-chain and off-chain approaches. On-chain strategies include sharding, consensus enhancements, and DAG-based architectures, while off-chain methods such as the Lightning Network and Raiden Network shift processing beyond the main blockchain. FortunChain, a proposed sharding-based blockchain, introduces the Tyche consensus protocol [Zhang, 23], which leverages a verifiable elliptic curve-based function for probabilistic and secure node selection. It integrates multi-period, fast, and dynamic sharding accounting to improve transaction processing, utilizing a verification chain for secondary validation and state sharding to maintain global transaction consistency.

A graph-based sharding scheme is introduced to optimize transaction distribution and minimize cross-shard transactions in public blockchains [Xu, 24]. Experimental results demonstrate a reduction in transaction confirmation latency and improved scalability. Another comprehensive sharding approach, OverShard, incorporates an overlapping network and virtual accounts to enhance security and inter-shard communication [Yu, 23]. OverShard enables nodes to operate across multiple shards, ensuring data availability via intra-shard communication. Its virtual accounts model,

built on a DAG structure, manages transactions and master accounts across shards, enabling near-linear scalability as the number of shards grows.

SPEX-Tran is a sharding-based framework designed to improve blockchain performance by enabling parallel execution of transactions within blocks, replacing traditional sequential processing [Chen, 23]. By utilizing sharding, SPEX-Tran minimizes execution and consensus overhead, encouraging nodes to actively verify transactions while enhancing security. An optimized parallel execution algorithm further boosts efficiency, with experimental results indicating significantly higher throughput than conventional sharding approaches.

Dioxide, a scalable Proof-of-Stake (PoS)-based sharding protocol, aims to mitigate the high orphan rate in Asynchronous Consensus Zones [Qu, 22]. It maintains both security and decentralization by employing a PoS consensus mechanism with fixed block intervals and a non-overlapping block producer selection strategy. A global beacon chain strengthens system security, and experimental evaluations demonstrate an increased average throughput compared to Asynchronous Consensus Zones.

EZchain introduces a novel consensus mechanism based on the value model, a transaction structure that integrates features of both account and UTXO models [Xue, 23]. By focusing on value transfer, EZchain enhances scalability, accelerates transaction confirmation, and optimizes storage efficiency, as verified through prototype simulations.

Tangle 2.0, a leaderless probabilistic consensus protocol, determines consensus based on the heaviest DAG rather than the longest chain, replacing PoW with a stake-or reputation-based weight function [Müller, 22]. This system, utilizing a Reality-based UTXO Ledger, facilitates parallel transaction validation without enforcing total ordering, enabling asynchronous ledger updates. Initial simulations confirm its improved performance.

High transaction latency and verification costs in existing DAG-based consensus protocols are addressed through a new commit rule grounded in the UTXO Data Model, which allows nodes to anticipate transaction outcomes before finalizing them, thereby reducing latency [Liu, 2023].

A DAG-based blockchain employing PoW consensus is proposed to enhance scalability while preserving security and decentralization [He, 21]. Key innovations include partitioning large blocks into smaller segments for better network efficiency, embedding a milestone chain to ensure security, and incorporating a transaction assignment method to reduce redundant processing by multiple miners. These improvements lead to faster consensus, higher transaction throughput (TPS), and lower latency while discouraging pool mining. Similarly, the proposed binary tree blockchain utilizes a structured hierarchy to distribute transactions efficiently across different paths, enabling parallel processing while maintaining transaction consistency, ultimately enhancing scalability without the need for complex conflict resolution mechanisms.

Phantasm extends the Phantom protocol [Sompolinsky, 20], introducing an adaptive, scalable mining technique to stabilize block ordering in DAG-based blockchains [Zhang, 2023]. It integrates two strategies for block referencing, aiding honest nodes and mitigating splitting attacks. Theoretical analysis and simulations confirm Phantasm's stability and improved efficiency over Phantom.

NEZHA, a concurrency control mechanism for DAG-based blockchains, is designed to enhance throughput and minimize processing latency by addressing

transaction conflicts [Xiao, 22]. It constructs an address-based conflict graph and employs a hierarchical sorting algorithm to establish a total order among transactions, thereby reducing transaction aborts. Experimental evaluations demonstrate NEZHA's superior performance compared to conventional conflict graph approaches, significantly improving throughput while reducing processing delays.

The Reality-based Ledger, an enhanced UTXO Ledger model, improves scalability by optimistically updating the ledger and managing conflict dependencies [Müller, 23]. It enables parallel transaction processing by maintaining multiple conflict-free ledger states. To facilitate this, the Branch DAG is introduced, efficiently updating ledger structures upon new transactions, minimizing delays, and supporting stream processing. This framework removes the need for centralized control and total ordering, which often limit scalability in traditional UTXO models.

HybridChain, a distributed ledger system, combines sharded blockchain and DAG technologies with a decentralized learning-based consensus algorithm [Taherpour, 24]. Validators exchange votes to resolve transaction conflicts and reach consensus. Comparisons with IOTA [Popov, 18] and Omniledger [Kokoris-Kogias, 18] show that IOTA achieves high throughput at the expense of accuracy, while Omniledger maintains accuracy but with higher latency. In contrast, HybridChain offers fast, accurate, and secure transactions with high scalability. It also strengthens security through decentralized storage and computation, ensuring low latency and consistent accuracy across different transaction rates.

AdaptChain, an adaptive scaling blockchain, adjusts throughput dynamically based on transaction demand [Xu, 23]. It expands under high transaction loads and contracts when demand decreases to optimize communication and storage costs. A transaction deduplication mechanism prevents duplicate transactions in parallel block additions, enhancing throughput and bandwidth efficiency. Additionally, AdaptChain includes a mining power load balancing mechanism to defend against attacks and operates in epochs, resizing the blockchain based on previous block sizes.

The Mutable Block with Immutable Transactions (MBIT) model addresses scalability and storage challenges in blockchain systems [Kottursamy, 23]. MBIT leverages a trapdoor cryptographic hash function to store user-specific transactions in blocks, allowing new transactions to be added without altering existing ones. This ensures immutability, consistency, and security, with experimental results showing faster verification and confirmation times compared to traditional approaches.

HuaBaseChain is developed to tackle blockchain inefficiencies in IoT networks [Mao, 23]. It reduces energy consumption, introduces a multidimensional digital token model, and lowers storage requirements. By implementing Proof-of-Participation (PoP) consensus, HuaBaseChain significantly enhances energy efficiency while extending the single-token model into a multidimensional system, enabling diverse IoT applications. Experiments confirm increased efficiency, lower storage needs, and faster query performance.

Red Belly Blockchain (RBBC) is a secure blockchain designed for large-scale geographically distributed consensus [Crain, 21]. It introduces three key innovations: (1) solving the Set Byzantine Consensus problem to reach consensus on a superblock of proposed blocks, (2) employing a leaderless design to resist censorship and ensure valid transactions are committed, and (3) implementing sharded verification to reduce signature verification overhead while maintaining security.

The Lightning Network and Raiden Network serve as off-chain scalability solutions that enhance blockchain performance by moving transaction processing off the main chain [Matani, 24]. These networks significantly improve scalability by reducing blockchain workload and storage requirements, enabling faster and more efficient transaction processing outside the primary blockchain ledger.

Although sharding-based and DAG-based solutions have enhanced blockchain scalability, they still encounter challenges such as transaction ordering complexity, security trade-offs, and cross-shard communication overhead. To address these issues, the proposed binary tree blockchain structure introduces several improvements. It enables efficient and deterministic transaction placement through binary addressing, ensuring a structured and predictable organization of data. Additionally, it enhances transaction immutability by decomposing transactions into smaller elements, which are distributed across different paths, reducing the risk of inconsistencies. The hierarchical nature of the system further minimizes computational overhead by optimizing transaction storage and retrieval processes. Moreover, the binary tree structure inherently provides an implicit form of sharding, allowing transactions to be automatically distributed and balanced across the network. By integrating these advancements, the proposed approach contributes to ongoing research in blockchain scalability and transaction efficiency, presenting a viable alternative to existing DAG-based and sharding-oriented frameworks.

3 The Proposed System Model

The proposed system model is based on a blockchain architecture that utilizes a binary tree structure to enhance scalability and efficiency in transaction processing. Among the various types of binary trees, such as B-trees and B+-trees, a binary tree data structure has been specifically chosen for its efficient node addressing capabilities. In a binary tree, nodes can be uniquely identified using binary paths (sequences of 0s and 1s), making the process of locating and addressing nodes computationally efficient. Unlike B-trees, where node addresses may change due to rebalancing, binary trees maintain consistent node addressing. Additionally, a binary tree inherently supports scalability through its hierarchical structure, which enables efficient workload distribution. Its design facilitates the distribution of transactions across different branches, effectively achieving sharding and allowing parallel transaction processing. However, the binary tree used in this system is unbalanced because of the randomness of used binary paths. This randomness causes transactions and their components to underutilize some branches of the binary tree, leading to irregularity. Nevertheless, the randomness ensures that the load is distributed proportionally according to the characteristics of the input data. Moreover, the binary tree structure is simpler to implement and computationally less demanding compared to more complex tree structures, while still fulfilling the requirements of the proposed system.

3.1 System Model Description

The model introduces a binary tree structure to blockchain design, wherein each node in the binary tree represents a block in the blockchain. The tree grows dynamically as

new blocks are added, enabling parallel processing and improved load distribution across the network.

The blockchain network consists of a set of nodes, each representing a participant in the system. These nodes are organized in a peer-to-peer (P2P) network, where each node can communicate directly with other nodes. The network follows a decentralized model, ensuring that no single entity controls the system. Nodes in the network can act as miners, validators, and participants in the transaction process. Each node maintains a local copy of the binary tree blockchain and participates in the consensus process.

The binary tree blockchain keeps transactions because the proposed method relies on UTXO model. In the UTXO model, money (coin) movements are realized using transactions. Each transaction consumes one or more previously unspent transaction outputs and transfers money to one or more new unspent outputs. This process creates chains between transactions while conveying money from accounts to accounts.

To preserve scalability of the binary tree, the transactions should be distributed properly over the binary tree nodes. In other words, the loads on each node should be nearly equal. To provide an equal distribution, a novel method is proposed. Each transaction is decomposed into its input and output elements. Each element is called a transaction element (trel). Each trel is associated with an account and is stored in the binary tree according to a specific blockchain path determined by the account's address.

The core of the system is a binary tree structure where each node in the tree represents a block in the blockchain. The tree grows by adding new blocks at the leaf nodes, with each block containing a set of transactions and trels. The root of the binary tree is the genesis block, which is the first block in the blockchain. It is embedded in the system at initialization and contains initial system parameters. Each block in the binary tree contains a set of transactions and trels, a reference to its parent block, and a unique identifier based on its position in the tree. The blocks are linked in a hierarchical manner, with each block having a path represented by a binary string of 0s and 1s.

In Bitcoin, a single chain of blocks is generated starting from the genesis block. The proposed method uses binary tree instead of a single chain. Each node in the binary tree contains a block in the blockchain. Figure 1 illustrates the binary tree of the blockchain, showing three layers. The root node is the genesis block of the blockchain. Each node has two child nodes, which are addressed as 0 and 1. Consequently, a node in the binary tree can be located uniquely with a path of 0s and 1s starting from the genesis block. Furthermore, a path from the root node to a leaf node can be addressed with a binary number sequence in the binary tree. These binary paths are called blockchain paths. From the genesis block, the binary tree can add two blocks, and likewise, for new layers in the binary tree, the number of blocks added will double each time. Therefore, binary trees are scalable data structures.

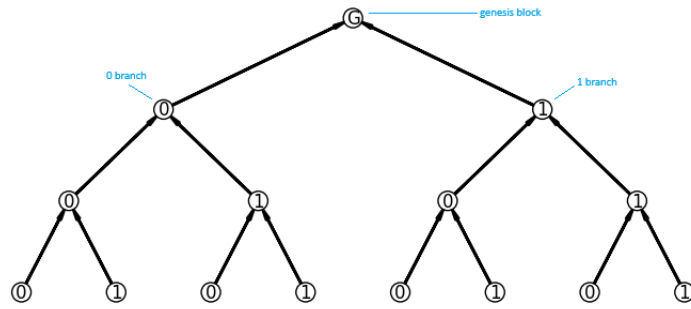


Figure 1: Binary tree nodes addressed with 0s and 1s

The system employs a Proof-of-Work (PoW) consensus mechanism, similar to that used in Bitcoin, to validate new blocks and maintain the integrity of the blockchain. Nodes compete to solve a cryptographic puzzle to add new blocks to the binary tree. The first node to solve the puzzle broadcasts the new block to the network, which is then verified and added to the blockchain by other nodes. Before adding a block to the blockchain, nodes verify that all transactions and trels within the block are valid and that the block itself follows the correct path in the binary tree.

A transaction can have two or more trels, which represent the inputs or outputs of a transaction. In Figure 2, a sample transaction with 2 inputs and 3 outputs is depicted without considering the payment of a fee. The letters in the transaction represent accounts. In other words, 10-unit coins from account A and 5-unit coins from account B are transferred to account C with 3-unit coins, to account D with 8-unit coins, and to account E with 4-unit coins.

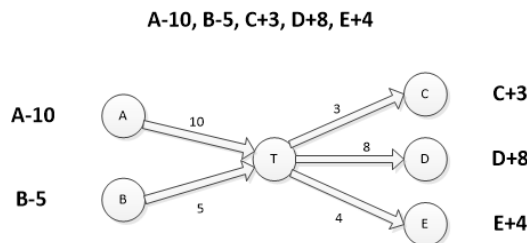


Figure 2: A sample transaction

In this sample transaction, there are 2 input trels and 3 output trels. Input trels can be shown with account name and minus sign, output trels can be shown with account name and plus sign. In other words, there are 5 trels. Each trel is directed to a specific blockchain path according to its account address. Thus, each trel will be located in a specific blockchain path where its account address is compatible with the blockchain path. For big binary trees, the account addresses may be shorter in length. For this reason, a repetition of the account address is used, and the blockchain path should be the starting part of it. In this manner, all the information about a specific account is

found in the related blockchain path. Consequently, the balance of an account can be calculated by summing the related trels in the related blockchain path.

Account addresses can be thought simply as public keys of the related private keys. Therefore, account addresses are random numbers; in this way, trels are distributed randomly over the binary tree blockchain.

The transactions are recorded in the binary tree blockchain after recording its trels. The transactions are also directed to specific blockchain paths according to their IDs using the same method as with trels. Transaction IDs are hashes of transaction data. Therefore, transactions are also randomly distributed over the binary tree blockchain. Each transaction has references to its trels. Thus, from a transaction all the trels can be found in pointed blocks in the related blockchain paths. Likewise, each trel also keeps its transaction ID. Therefore, from a trel the related transaction can be found in the specific blockchain path.

The proposed method generates a binary tree blockchain with programs running in nodes in the blockchain network, which are called blockchain node programs. Nodes can also be called miners without considering whether they create money or not. Nodes in the blockchain construct a network in a P2P manner. The genesis block, the first block of the blockchain, is embedded in the blockchain node programs. Each blockchain node program builds its own local blockchain starting from the genesis block.

The binary tree structure inherently supports scalability through implicit sharding. Transactions are distributed across different paths in the tree, allowing for parallel processing and reducing bottlenecks. The binary tree structure divides the blockchain into multiple paths, each representing a shard. Transactions are routed to specific shards based on their trels, enabling the system to process multiple transactions simultaneously without conflict. The decomposition of transactions into trels, representing the inputs or outputs of a transaction, and their distribution across the tree ensures that the computational load is evenly spread across the network, preventing any single node or path from becoming a bottleneck.

Security in the proposed system is achieved through the cryptographic properties of the PoW consensus mechanism and the hierarchical structure of the binary tree. Once a transaction is recorded in the blockchain, it becomes immutable due to the cryptographic linking of blocks. Each transaction indirectly verifies multiple paths in the binary tree, enhancing the overall security and consistency of the blockchain.

The decentralized nature of the network and the distribution of transactions across multiple paths reduce the risk of centralized attacks, such as double-spending or Sybil attacks.

The proposed system model is based on several key assumptions:

- **Network Reliability:** The P2P network is assumed to be reliable, with nodes able to communicate effectively and propagate information across the network in a timely manner.
- **Node Participation:** A sufficient number of nodes are assumed to participate in the consensus process, ensuring the security and decentralization of the blockchain.
- **Cryptographic Security:** The security of the system relies on the assumption that the cryptographic algorithms used in the PoW mechanism are secure and resistant to known attacks.

- While the binary tree blockchain offers significant scalability benefits, it may face challenges in environments with highly variable network conditions or limited node participation.

3.2 Algorithm Description

The algorithm underlying the binary tree blockchain is designed to maximize scalability while maintaining the security and efficiency of the transaction processing. Algorithm 1 shows the steps of decomposition of transactions. Each transaction submitted to the network is decomposed into smaller elements (trels). Each trel corresponds to an input or output in the transaction and is assigned a unique identifier. Each transaction is hashed and assigned to a unique identifier.

Algorithm 1 Transaction Element Decomposition

Input: A transaction T containing multiple inputs and outputs.

Output: A set of transaction elements (trels).

- 1: For each input in T, create an input trel and assign it a unique identifier.
- 2: For each output in T, create an output trel and assign it a unique identifier.
- 3: Store the input and output trels and transactions in a set Trels(T).

Comment: Each input and output is converted into trels, ensuring they can be independently processed and stored in the binary tree blockchain.

Algorithm 2 shows the steps of assignment of binary paths, which represent the addresses of nodes in the binary tree. Transactions are distributed across the binary tree blockchain according to their identifiers, and trels are distributed across the binary tree blockchain according to their associated account addresses. The binary path to store each transaction is determined by the transaction identifier, and the binary path to store each trel is determined by the account address.

Algorithm 2 Binary Path Assignment

Input: A set of transactions and trels Trels(T).

Output: A set of binary paths for each transaction and trel.

- 1: For each trel in Trels(T)
 - repeat
 - the account address of the trel is concatenated to the binary string
 - until the length of the binary string is sufficient to address the node position
- 2: For each transaction in Trels(T)
 - repeat
 - the transaction hash is concatenated to the binary string
 - until the length of the binary string is sufficient to address the node position
- 3: For each trel or transaction in Trels(T), assign to a binary path in the binary tree based on the binary string.

Comment: An account address is a bit string. This bit string is concatenated to itself as many times as necessary to obtain the related binary path. Similarly, a transaction hash is also a bit

string. The concatenation of it to itself as many times as necessary produces the related binary path for the transaction. Binary paths ensure a consistent and predictable way to place data in the binary tree blockchain.

Algorithm 3 shows the steps of block formation. Blocks are formed by aggregating a sufficient number of transactions and trels. Each block is assigned a position in the binary tree and contains references to its parent block, related unspent transactions, and its related trels.

Algorithm 3 Block Formation

Input: A set of transactions and trels with assigned binary paths.

Output: A block B in the binary tree.

- 1: Aggregate a sufficient number of transactions and trels to form a block.
- 2: Assign block B a position in the binary tree based on the binary paths of its transactions and trels.
- 3: Include references to the parent block, related transactions and trels.

Comment: Each block contains references to its parent and the trels/transactions it stores, ensuring hierarchical linkage and efficient data retrieval.

Algorithm 4 shows the steps of mining and consensus. The mining process involves solving a cryptographic puzzle to generate a new block. Once a block is mined, it is broadcasted to the network, where other nodes validate its contents and add it to their local copy of the binary tree.

Algorithm 4 Mining and Consensus

Input: A new block B.

Output: A validated block added to the blockchain.

- 1: Nodes compete to solve a cryptographic puzzle related to B.
- 2: The first node to solve the puzzle broadcasts B to the network.
- 3: Other nodes validate B by checking the correctness of its transactions and trels and its binary path.
- 4: If B is valid, add it to the binary tree blockchain.

Comment: This ensures that only valid blocks are added to the binary tree blockchain, maintaining its integrity and consistency.

Algorithm 5 shows the steps of validation and propagation. Nodes validate each received block by checking the correctness of its transactions and trels and ensuring that the block follows the correct binary path (the address of a node in the binary tree). Valid blocks are propagated across the network.

Algorithm 5 Validation and Propagation

Input: A newly mined block B.

Output: Block B is added to the blockchain and propagated across the network.

1: Upon receiving B, each node verifies its validity by checking the correctness of its transactions and trels.

2: If valid, the node adds B to its local copy of the blockchain.

3: Propagate B to other nodes in the network.

Comment: This process ensures that all nodes in the network maintain a consistent view of the blockchain and that invalid blocks are not propagated.

3.3 Correctness

The correctness of the proposed algorithm is based on its adherence to the rules of the binary tree structure and the PoW consensus mechanism. The following properties ensure the correctness of the algorithm:

- The binary tree structure is maintained by ensuring that each block is added to the correct position in the tree, based on its binary path (the address of a node in the binary tree) according to the contained transactions and trels (the inputs or outputs of transactions). The integrity of the tree is preserved by the cryptographic linking of blocks.
- Each transaction and trel is validated before being added to the blockchain. The validation process ensures that no double-spending occurs, and that each transaction adheres to the UTXO model.
- The PoW consensus mechanism ensures that all nodes agree on the state of the blockchain. The cryptographic puzzle prevents the creation of conflicting blocks, maintaining the consistency of the blockchain.

4 Experimental Results

The proposed novel method is simulated with a Python program. The nodes in the blockchain network are simulated with threads. There are 100 threads. In other words, there are 100 miners. This selection reflects a moderately sized network suitable for experimental setups, effectively balancing computational resources. It provides an adequate representation of the distributed nature of the blockchain and captures a representative sample of mining competition. Additionally, this scale aligns with similar studies in blockchain research, offering valuable insights into scalability and performance without overburdening the simulation. In Bitcoin, a block is generated approximately every 10 minutes. For the simulation, a block generation time of 10 seconds is used. The fixed block generation time is a pragmatic choice for the simulation, enabling manageable testing cycles while preserving an appropriate difficulty level for PoW systems. It represents an accelerated model compared to Bitcoin's 10-minute interval, facilitating the evaluation of performance over shorter timescales.

The proposed binary tree blockchain can be implemented with different consensus algorithms. However, PoW used in Bitcoin is selected for the simulation. Therefore, a

constant block generation time, which represents validation time in PoW, is determined as 10 seconds. In the simulation, 1000 blocks are generated. Therefore, a Bitcoin-like blockchain can generate 1000 blocks in 10000 minutes.

In the simulation, random transactions are created, which can have at least 2 trels (the inputs or outputs of transactions), with one of them being an input trel and the other an output trel. Additionally, a random transaction can have up to 6 trels. Blocks are added to the binary tree blockchain if they have at least 5 transactions or trels, and at most nearly 10 transactions or trels.

Blocks	Time (s)	Average Time (s)	Cumulative Time (s)	Average Cumulative Time (s)
0 - 99	23,51	0,24	23,51	0,24
100 - 199	17,39	0,17	40,90	0,20
200 - 299	10,37	0,10	51,27	0,17
300 - 399	10,55	0,11	61,82	0,15
400 - 499	9,71	0,10	71,53	0,14
500 - 599	10,50	0,11	82,03	0,14
600 - 699	13,23	0,13	95,27	0,14
700 - 799	9,27	0,09	104,54	0,13
800 - 899	19,11	0,19	123,65	0,14
900 - 999	10,92	0,11	134,56	0,13

Table 1: Block generation times

In Table 1, the simulation times are depicted. Every 100 blocks generation is taken as a unit. The second column shows the generation time for each unit. The first 100 blocks are generated in 23.51 seconds, and the last 100 blocks are generated in 10.92 seconds. In the third column, the average generation time is shown for each unit. The average block generation time for the first 100 blocks is 0.24 seconds, and the average block generation time for the last 100 blocks is 0.11 seconds. The fourth column shows the cumulative generation times. The first 500 blocks are generated in 71.53 seconds, and all the blocks are generated in 134.56 seconds. The last column shows the average generation time for cumulative units. The average generation time of a block is 0.14 seconds for the first 500 blocks, and the average block generation time is 0.13 seconds for all the blocks.

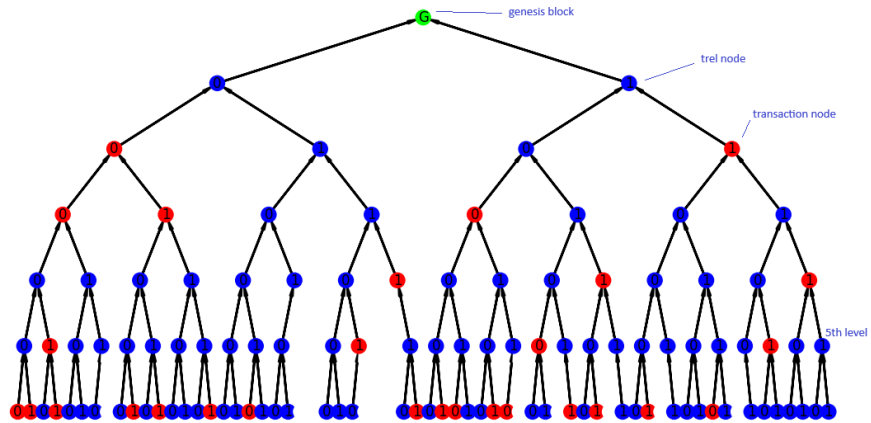


Figure 3: Binary tree blockchain simulation

In Figure 3, the binary tree blockchain simulation is shown. The simulation produced the binary tree up to 22 levels from the root block (the genesis block). In the figure, only 6 levels are shown. Each node in the tree is depicted with a binary number 0 or 1 according to its position. In the 5th level, two nodes have not been generated because there is no suitable transaction or trel (transaction element) for those blockchain paths. In the next level, 9 blocks have not been generated also. The nodes with blue color show the blocks which only include trels. These nodes are called trel nodes. Therefore, the first level has blue nodes, since the transactions can be recorded after recording the related trels. The nodes with red color show the blocks which include at least one transaction. These nodes are called transaction nodes.

Miners	0	1	2	3	4	5	6	7	8	9
0+	13	13	12	11	11	10	12	9	12	8
10+	11	8	10	11	11	9	11	11	11	12
20+	9	11	10	11	12	11	11	10	10	11
30+	10	11	10	9	6	10	10	10	11	11
40+	10	8	11	10	11	11	10	10	10	10
50+	11	8	11	10	10	8	10	10	9	8
60+	9	10	8	10	10	10	8	10	10	9
70+	10	10	10	10	10	10	10	9	9	10
80+	10	8	10	10	9	8	10	10	8	10
90+	9	10	12	6	10	10	10	10	11	11

Table 2: Miner block generation counts

In Table 2, the block generation counts are shown for all the 100 miners. In the first column, the base number is given for a miner ID. In the header of columns, an additional number is given for a miner ID. In other words, the sum of the base and additional numbers shows the miner ID of the specific cell. The average of miner block generation counts is 10. Moreover, the standard deviation of the counts is 1.23. This relatively small standard deviation indicates low variability in block generation, suggesting that the workload is distributed fairly among the miners. Therefore, all the miners generated nearly 10 blocks for the binary tree blockchain.

	Average	Standard Deviation
Level	15.67	1.33
Transaction node count	7.09	1.84
First transaction node level count	3.75	2.63
Last transaction node level count	15.56	1.37
Transaction node count per level	0.45	0.11

Table 3: Leaf counts distribution

The binary tree blockchain simulation generated up to 22 levels in the binary tree. There are 99 leaves in the tree. The analysis of the tree is given in Table 3. The first row in the table is about leaf levels. The average leaf levels are 15.67, and the standard deviation is 1.13. In other words, the leaves in the tree are in approximately level 16. The second row in the table is about transaction nodes. Ninety-nine leaves mean that there are 99 blockchain paths. The average transaction node count in each blockchain path is 7.09, and standard deviation is 1.84. Therefore, approximately half of the nodes in each blockchain path are transaction nodes. Transaction nodes are important because they approve other blockchain paths via their trels, which represent the inputs or outputs of a transaction. This increases the immutability of the blockchain. The third row in the table is about transaction node positions in each blockchain path. The first transaction node level means the first level of transaction nodes in a blockchain path. The average first transaction node level count is 3.75, and standard deviation is 2.63. In other words, nearly at level 4 are transaction nodes in the tree. The last transaction node level means the last level of transaction nodes in a blockchain path. The average last transaction node level count is 15.56, and standard deviation is 1.37. In other words, nearly at level 16 in the binary tree are transaction nodes for the 22-level binary tree blockchain. The last row in the table summarizes the characteristics of the blockchain. The average transaction node count per level is 0.45, and standard deviation is 0.11. In brief, half of the nodes in the blockchain are transaction nodes.

5 Discussion

In the proposed method, each transaction is decomposed into its trels (the inputs or outputs of transactions) to distribute transactions over the binary tree. Each trel is directed to a specific blockchain path determined by its account address, which ensures that trels are consistently located within the same path. This methodology enhances the

blockchain's scalability by evenly distributing the processing load and minimizing the potential for bottlenecks.

One significant advantage of this approach is the increase in blockchain immutability. Transactions can only be recorded in the binary tree blockchain if their corresponding trels have already been recorded. As a result, each transaction indirectly verifies multiple parts of the binary tree, thereby enhancing the overall consistency and reliability of the blockchain. This feature is particularly beneficial for applications requiring high levels of data integrity and security.

The method also provides an implicit form of sharding. By using a binary tree structure to organize and manage transactions, the blockchain can handle a higher volume of transactions in parallel, which is a key requirement for scalable blockchain systems. This implicit sharding, which refers to the automatic distribution of transactions or trels across different parts of the binary tree, reduces the computational overhead and allows for faster transaction processing and verification compared to traditional blockchain architectures.

	IOTA	Nano	Avalanche	Hedera	Fantom	Bitcoin	The Proposed Blockchain
TPS	1000	7000	160	2500	476	7	32307
Latency (seconds)	100	10	8	25	6	600	1200

Table 4: Comparison of DAG-based blockchain technologies

Table 4 compares major DAG-based blockchain technologies based on their TPS and latency values [Kahmann, 23] [Li, 21] [Göbel, 17]. While Bitcoin is not a DAG-based system, it is included as a benchmark due to its importance as a representative of traditional blockchain technologies and because the proposed methodology is directly compared to Bitcoin. With a TPS of 7 and a latency of 10 minutes (600 seconds), Bitcoin highlights the limitations of conventional blockchain designs in terms of throughput and latency. In contrast, DAG-based blockchains like Nano, Hedera, and Fantom show significant advancements, achieving higher TPS and lower latency through their parallelized transaction validation and consensus mechanisms. The proposed blockchain methodology enhances scalability by generating blocks every 0.13 seconds on average, with a block generation time of 10 seconds. This approach uses a two-stage confirmation process: first, adding trels to the blockchain, and then incorporating the actual transactions. As a result, the latency for the proposed system is 20 seconds, which is twice the block generation time. When scaled to match Bitcoin's TPS and latency values, the proposed blockchain achieves a remarkable TPS of 32307 and a latency of 1200 seconds. Compared to the major DAG-based blockchains, which average approximately 2227 TPS and 29 seconds latency, the proposed blockchain offers a significantly higher throughput but with an increase in latency.

Moreover, the use of the UTXO model for transaction distribution further contributes to scalability. The UTXO model's inherent design supports parallel processing of transactions, as each transaction can be processed independently. This reduces the complexity and computational load associated with transaction verification and balance calculations.

The experimental results underscore the effectiveness of the proposed method. The binary tree blockchain demonstrated the ability to handle a significantly higher transaction throughput than traditional blockchain systems. The simulation results showed that block generation times decreased as more blocks were added, and the workload was evenly distributed among the miners, indicating a well-balanced system. This approach offers a promising direction for future research and development in blockchain technology, addressing the critical challenges that currently hinder its widespread adoption.

6 Conclusion

The widespread adoption of blockchain technologies hinges on addressing scalability challenges. This paper proposes a novel approach utilizing a binary tree data structure to enhance blockchain scalability. By distributing transactions across the binary tree, this method achieves several advantages.

Efficient node addressing is achieved by using binary numbers, providing a straightforward and effective method for identifying and locating nodes. Each node in the tree contains a block of transactions, which are directed to specific paths within the tree. This directed path enhances scalability by enabling parallel processing of transactions, ensuring that the blockchain can handle a higher volume of transactions without becoming congested.

Transactions are decomposed into transaction elements, which improves the immutability of the binary tree blockchain and minimizes balance calculations. This decomposition process reduces computational overhead, allowing for rapid and accurate transaction processing and verification. The binary tree structure effectively realizes implicit sharding, distributing the processing load more evenly and reducing bottlenecks. This results in a more scalable and efficient blockchain system.

In conclusion, the proposed method of using a binary tree data structure for blockchain scalability offers a promising solution to the blockchain trilemma. By leveraging the inherent properties of binary trees and incorporating transaction element decomposition, this approach achieves significant improvements in scalability. Continuous development and exploration of advanced solutions are essential for overcoming the scalability challenges that hinder the widespread adoption of blockchain technologies.

Data Availability Statement

The data underpinning the analysis reported in this paper are deposited at “Data repository” Zenodo at <https://doi.org/10.5281/zenodo.14845656>.

References

[Abbas, 22] Abbas, H., Caprolu, M., & Di Pietro, R. (2022, August). Analysis of polkadot: Architecture, internals, and contradictions. In 2022 IEEE International Conference on Blockchain (Blockchain) (pp. 61-70). IEEE.

[Bachani, 22] Bachani, V., & Bhattacharjya, A. (2022). Preferential delegated proof of stake (PDPoS)—modified DPoS with two layers towards scalability and higher TPS. *Symmetry*, 15(1), 4.

- [Baird, 16] Baird, L. (2016). The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep, 34, 9-11.
- [Berger, 23] Berger, C., Schwarz-Rüsch, S., Vogel, A., Bleeke, K., Jehl, L., Reiser, H. P., & Kapitza, R. (2023, May). Sok: Scalability techniques for BFT consensus. In 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-18). IEEE.
- [Buterin, 13] Buterin, V. (2013). Ethereum white paper. GitHub repository, 1, 22-23.
- [Chen, 23] Chen, G., Zhang, J., Wu, W., & Zhou, J. (2023, May). Parallel Execution of Blockchain Transactions with Sharding. In ICC 2023-IEEE International Conference on Communications (pp. 6559-6564). IEEE.
- [Crain, 21] Crain, T., Natoli, C., & Gramoli, V. (2021, May). Red belly: A secure, fair and scalable open blockchain. In 2021 IEEE Symposium on Security and Privacy (SP) (pp. 466-483). IEEE.
- [Göbel, 17] Göbel, J., & Krzesinski, A. E. (2017, November). Increased block size and Bitcoin blockchain dynamics. In 2017 27th International Telecommunication Networks and Applications Conference (ITNAC) (pp. 1-6). IEEE.
- [He, 21] He, J., Wang, G., Zhang, G., & Zhang, J. (2021). Consensus mechanism design based on structured directed acyclic graphs. *Blockchain: Research and Applications*, 2(1), 100011.
- [Jourenko, 22] Jourenko, M., Larangeira, M., & Tanaka, K. (2022, July). Interhead hydra: Two heads are better than one. In *The International Conference on Mathematical Research for Blockchain Economy* (pp. 187-212). Cham: Springer International Publishing.
- [Kahmann, 23] Kahmann, F., Honecker, F., Dreyer, J., Fischer, M., & Tönjes, R. (2023). Performance Comparison of Directed Acyclic Graph-Based Distributed Ledgers and Blockchain Platforms. *Computers*, 12(12), 257.
- [Khan, 21] Khan, D., Jung, L. T., & Hashmani, M. A. (2021). Systematic literature review of challenges in blockchain scalability. *Applied Sciences*, 11(20), 9372.
- [Kokoris-Kogias, 18] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018, May). Omniledger: A secure, scale-out, decentralized ledger via sharding. In 2018 IEEE symposium on security and privacy (SP) (pp. 583-598). IEEE.
- [Kottursamy, 23] Kottursamy, K., Sadayapillai, B., AlZubi, A. A., & Bashir, A. K. (2023). A novel blockchain architecture with mutable block and immutable transactions for enhanced scalability. *Sustainable Energy Technologies and Assessments*, 58, 103320.
- [Lamport, 01] Lamport, L. (2001). Paxos made simple. *ACM SIGACT News (Distributed Computing Column)* 32, 4 (Whole Number 121, December 2001), 51-58.
- [Li, 21] Li, L., Shi, P., Fu, X., Chen, P., Zhong, T., & Kong, J. (2021). Three-dimensional tradeoffs for consensus algorithms: A review. *IEEE Transactions on Network and Service Management*, 19(2), 1216-1228.
- [Li, 23] Li, Y., Wang, J., & Zhang, H. (2023). A survey of state-of-the-art sharding blockchains: Models, components, and attack surfaces. *Journal of Network and Computer Applications*, 103686.
- [Liu, 2023] Liu, K., Jourenko, M., & Larangeira, M. (2023). Reducing Latency of DAG-based Consensus in the Asynchronous Setting via the UTXO Model. arXiv preprint arXiv:2307.15269.
- [Liu, 22] Liu, Y., Liu, J., Salles, M. A. V., Zhang, Z., Li, T., Hu, B., ... & Lu, R. (2022). Building blocks of sharding blockchain systems: Concepts, approaches, and open problems. *Computer Science Review*, 46, 100513.

- [Liu, 23] Liu, X., Xie, H., Yan, Z., & Liang, X. (2023). A survey on blockchain sharding. *ISA transactions*, 141, 30-43.
- [Luu, 16] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016, October). A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 17-30).
- [Mao, 23] Mao, X., Li, C., Zhang, Y., Zhang, G., Li, J., Shah, M., & Xing, C. (2023). HuaBaseChain: an extensible blockchain with high performance. *IEEE Internet of Things Journal*.
- [Matani, 24] Matani, A., Sahafi, A., & Broumandnia, A. (2024). A Comprehensive Review on Blockchain Scalability. *Journal of Electrical and Computer Engineering Innovations (JECEI)*, 12(1), 187-216.
- [Monrat, 23] Monrat, A. A., Schelén, O., & Andersson, K. (2023, July). Addressing the Performance of Blockchain by Discussing Sharding Techniques. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)* (pp. 1-9). IEEE.
- [Müller, 22] Müller, S., Penzkofer, A., Polyanskii, N., Theis, J., Sanders, W., & Moog, H. (2022). Tangle 2.0 leaderless nakamoto consensus on the heaviest dag. *IEEE Access*, 10, 105807-105842.
- [Müller, 23] Müller, S., Penzkofer, A., Polyanskii, N., Theis, J., Sanders, W., & Moog, H. (2023). Reality-based UTXO ledger. *Distributed Ledger Technologies: Research and Practice*, 2(3), 1-33.
- [Nakamoto, 08] Nakamoto, S. (2008). Bitcoin. A peer-to-peer electronic cash system, 21260.
- [Nasir, 22] Nasir, M. H., Arshad, J., Khan, M. M., Fatima, M., Salah, K., & Jayaraman, R. (2022). Scalable blockchains—A systematic review. *Future generation computer systems*, 126, 136-162.
- [Ongaro, 14] Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference (USENIX ATC 14)* (pp. 305-319).
- [Paulavičius, 2021] Paulavičius, R., Grigaitis, S., & Filatovas, E. (2021). A systematic review and empirical analysis of blockchain simulators. *IEEE access*, 9, 38010-38028.
- [Paulavičius, 21] Paulavičius, R., Grigaitis, S., & Filatovas, E. (2021, May). An overview and current status of blockchain simulators. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 1-3). IEEE.
- [Popov, 18] Popov, S. (2018). The tangle. *White paper*, 1(3), 30.
- [Qu, 22] Qu, C., Xiong, H., Wang, S., Niu, Y., & Li, J. (2022, April). Dioxide: A Proof-of-Stake Blockchain Consensus on Asynchronous Consensus Zones. In *2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)* (pp. 1-6). IEEE.
- [Sanka, 21] Sanka, A. I., & Cheung, R. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications*, 195, 103232.
- [Singh, 22] Singh, A., Kumar, G., Saha, R., Conti, M., Alazab, M., & Thomas, R. (2022). A survey and taxonomy of consensus protocols for blockchains. *Journal of Systems Architecture*, 127, 102503.
- [Sompolinsky, 20] Sompolinsky, Y., Wyborski, S., & Zohar, A. (2020). PHANTOM and GHOSTDAG, A Scalable Generalization of Nakamoto Consensus.(2020).

- [Taherpour, 24] Taherpour, A., & Wang, X. (2024). HybridChain: Fast, Accurate, and Secure Transaction Processing With Distributed Learning. *IEEE Transactions on Parallel and Distributed Systems*.
- [Wang, 19] Wang, J., & Wang, H. (2019). Monoxide: Scale out blockchains with asynchronous consensus zones. In *16th USENIX symposium on networked systems design and implementation (NSDI 19)* (pp. 95-112).
- [Wu, 22] Wu, H. Y., Yang, X., Yue, C., Paik, H. Y., & Kanhere, S. S. (2022). Chain or DAG? Underlying data structures, architectures, topologies and consensus in distributed ledger technology: A review, taxonomy and research issues. *Journal of Systems Architecture*, 131, 102720.
- [Xiao, 22] Xiao, J., Zhang, S., Zhang, Z., Li, B., Dai, X., & Jin, H. (2022, July). Nezha: Exploiting concurrency for transaction processing in dag-based blockchains. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)* (pp. 269-279). IEEE.
- [Xu, 18] Xu, M., Guo, Y., Liu, C., Hu, Q., Yu, D., Xiong, Z., ... & Cheng, X. (2018). Exploring blockchain technology through a modular lens: A survey. *ACM Computing Surveys*.
- [Xu, 23] Xu, J., Xie, Q., Peng, S., Wang, C., & Jia, X. (2023). Adaptchain: Adaptive scaling blockchain with transaction deduplication. *IEEE Transactions on Parallel and Distributed Systems*.
- [Xu, 24] Xu, S., Wang, Z., Wang, L., Mihaljević, M. J., Zhang, S., Shao, W., & Wang, Q. (2024). A Sharding Scheme Based on Graph Partitioning Algorithm for Public Blockchain. *CMES-Computer Modeling in Engineering and Sciences*.
- [Xue, 23] Xue, L., Yang, W., & Li, W. (2023). A Scale-out Decentralized Blockchain Ledger System for Web3. 0. arXiv preprint arXiv:2312.00281.
- [Yu, 23] Yu, B., Zhao, H., Zhou, T., Sheng, N., Li, X., & Xu, J. (2023). OverShard: Scaling blockchain by full sharding with overlapping network and virtual accounts. *Journal of Network and Computer Applications*, 220, 103748.
- [Zamani, 18] Zamani, M., Movahedi, M., & Raykova, M. (2018, October). Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (pp. 931-948).
- [Zhang, 2023] Zhang, Z., Liu, X., Feng, K., Wan, M., Li, M., Dong, J., & Zhu, L. (2023). Phantasm: Adaptive Scalable Mining Toward Stable BlockDAG. *IEEE Transactions on Services Computing*.
- [Zhang, 23] Zhang, Q., Wang, S., Zhang, D., Wang, J., & Sun, J. (2023). FortunChain: EC-VRF-based scalable blockchain system for realizing state sharding. *IEEE Transactions on Network and Service Management*.