DOI: 10.37943/16CGOY7609

Akylbek Tokhmetov
Candidate of physical and mathematical Sciences, Associate Professor of the Department of Information Systems
attohmetov@mail.ru, orcid.org/0000-0003-0764-8574
L.N. Gumilyov Eurasian National University, Kazakhstan

Vyacheslav Lee
Master student of the Department of Information Systems
lee_v23@mail.ru, orcid.org/0009-0006-4202-8302
L.N. Gumilyov Eurasian National University, Kazakhstan

Liliya Tanchenko
Master of technical sciences, Department of Information Systems
ltanchenko@mail.ru, orcid.org/0000-0002-6811-2303
L.N. Gumilyov Eurasian National University, Kazakhstan

DEVELOPMENT OF DAG BLOCKCHAIN MODEL

Abstract: In this study the authors present an innovative approach to resolving scalability and efficiency challenges in blockchain technology through the integration of Directed Acyclic Graphs (DAGs). This approach helps to overcome the limitations of traditional blockchain systems, particularly in transaction processing. The classic blockchain has some problems as slow transaction processing and poor scalability. The authors offer Directed Acyclic Graph (DAG) as a scalable and energy-efficient alternative. The paper outlines the development of a DAG-based blockchain model, utilizing Python and Flask alongside the Ed25519 cryptographic curve. It conducts a comparative analysis of DAG with traditional consensus mechanisms like Proof of Work and Proof of Stake, underscoring the efficiency and security benefits of employment of DAG. The research methodology includes an extensive literature review and the construction of a practical model to demonstrate DAG's applicability in blockchain networks. Particularly notable is the exploration of DAG's potential in Internet of Things (IoT) ecosystems, addressing critical issues such as energy inefficiency and network communication challenges in existing consensus algorithms. The authors calculated the performance of the model and compared it with similar models on several evaluation criteria. The simulation results of our proposed model show an improvement in performance and security by minimizing end-to-end delay, time cost, energy consumption, and throughput. The model eliminates the limitations of classic blockchain systems, such as high latency and low scalability. It structures transactions and blocks as a DAG, which provides fast validation and high scalability without compromising security. The research demonstrates the transformative implications of DAG for advancing blockchain technology.

Key words: Blockchain Scalability, Blockchain Modeling, Directed Acyclic Graph, Consensus Mechanisms, Secure Data Management

Introduction

In recent years, blockchain technology has revolutionized various sectors, yet it faces a significant bottleneck in scalability and efficiency. This article proposes an innovative approach to overcoming these challenges by integrating DAGs into blockchain networks. DAGs, unlike

Received: 22.09.2023 Accepted 26.10.2023 Published 14.11.2023

traditional blockchain structures, do not require blocks to be added in a linear sequence. This unique structure presents an opportunity to significantly enhance the throughput and efficiency of blockchain systems.

The core of this discussion revolves around DAGs. A DAG is a structured graph characterized by the absence of directed cycles. It comprises vertices and edges, with each edge directing from one vertex to another, ensuring no possibility of returning to the same vertex via a sequence of directed edges. In blockchain context, this translates to a non-linear structure where transactions are interlinked with multiple previous transactions, forming a directed, acyclic graph.

In this study, DAGs are employed to address key limitations inherent in traditional blockchain networks, particularly those related to scalability, speed, and efficiency. By facilitating the concurrent addition of multiple transactions and interlinking them with several existing transactions, DAGs considerably enhance the transaction processing capacity of blockchain networks. This is particularly advantageous in scenarios demanding high transaction throughput, such as in IoT and financial services sectors.

The integration of DAGs into blockchain networks offers multiple benefits. Primarily, it diminishes the reliance on intensive proof-of-work consensus mechanisms, thereby reducing energy consumption and augmenting transactional speed. Furthermore, DAGs exhibit superior scalability in comparison to traditional blockchains, as they enable parallel processing of transactions. This attribute significantly increases the volume of transactions processed within the same time frame. Lastly, the structural composition of DAGs inherently bolsters security and provides robust resistance against certain types of cyber-attacks, due to the complex interconnectivity of transaction histories.

Following this introduction to DAGs and their potential applications in blockchain networks, the next section of this article will delve into a comprehensive review of existing literature. This review will critically analyze and discuss scholarly works on DAGs in blockchain, providing a broader perspective on the current state of research, emerging trends, and how these align with the innovative approach presented in this work.

Importantly, any effort to scale up capacity must navigate the delicate balance between enhancing throughput and upholding the fundamental tenets of security and decentralization within blockchain networks [1]. Blockchain is like a special book that helps people who don't trust each other to exchange information and data without needing some responsible intermediary. It keeps the information safe and private, and it works really fast and well. But sometimes, there can be a problem with making it work for a lot of people at the same time because it can slow down and use up a lot of space. [2]

Beyond capacity constraints, current blockchain systems grapple with a suite of pressing issues. One such challenge revolves around DAGs, which introduce a "doubly-complex" dimension to blockchain systems. The complexity arises from two facets: firstly, the intricate order relationships inherent in the DAG structure, and secondly, the presence of "missing information," signifying relationships in the partially ordered set (poset) that do not manifest as explicit edges within the DAG. This article delves into characterizing the mesoscopic structures within DAGs, shedding light on the intricate interplay between observed and unobserved transitive edges within the underlying poset [3].

Efforts to expedite user validation within blockchain networks have led to the emergence of DAGs as a promising solution. By constructing a DAG in the blockchain, it becomes possible to accommodate a substantially larger volume of transactions, making it well-suited for the demands of larger-scale network environments. However, this adoption of DAG-based blockchains necessitates a careful evaluation of the trade-off between security and scalability, especially within the context of Software-Defined Networking (SDN) and the Internet of Things

(IoT). This article explores the implications of employing DAG-based blockchain technology within the SDN-IoT landscape, shedding light on the intricate interplay between security and scalability considerations [4].

In the realm of blockchain, the consensus mechanism stands as a linchpin, ensuring the security and legitimacy of data recorded within the blocks. Various consensus mechanisms have emerged, each with its unique characteristics and strengths. Yet, there exists a conspicuous absence of comprehensive technical analysis and comparisons to guide the selection of an appropriate consensus mechanism for specific scenarios or applications. In response to this need, this article conducts a rigorous investigation into three mainstream consensus mechanisms within the blockchain space: Proof of Work (PoW), Proof of Stake (PoS), and Direct Acyclic Graph (DAG). The study evaluates their performance across key metrics, including the average time required to generate a new block, confirmation delay, Transaction Per Second (TPS), and confirmation failure probability. Notably, the findings illuminate the multifaceted factors that influence the consensus process, encompassing network resource considerations such as computation power, coin age, and buffer size, as well as the influence of varying network load conditions. The results reveal distinct sensitivities among these consensus mechanisms, with PoW and PoS reacting more to changes in network resources, while DAG exhibits heightened sensitivity to network load conditions [5].

In the pages that follow, this article comprehensively explores DAGs in the context of blockchain technology, shedding light on their intricacies, potential, and implications. It offers valuable insights into the intricate dynamics of DAG-based blockchains and the critical role they play in addressing the capacity limitations while preserving the essential attributes of security and decentralization.

In response to these challenges, the proposed work embarks on a pioneering journey to establish an energy-efficient framework underpinned by DAGs as a cornerstone of the blockchain architecture [6]. One of the pivotal breakthroughs in this arena centers on the development of an off-chain block that addresses critical concerns surrounding reliable outsourced computations. This block aims to resolve the efficient and secure generation of computations, while simultaneously ensuring accountability in the verification process. Within this intricate structure, a DAG becomes the bedrock upon which transactions of computation results and verification reports are meticulously recorded in a fully decentralized manner. The cryptographic hash of this block is securely etched onto the blockchain, bolstering the integrity and reliability of the system. Moreover, the fusion of off-chain verification and on-chain arbitration delivers a robust mechanism for verification, further reinforced by a trust evaluation model that instills accountability among edge nodes [7].

In a broader context, blockchain technology promises a transformative approach to data sharing and collaboration among Internet of Things (IoT) devices, particularly when centralized IT infrastructure is unavailable. However, the existing consensus algorithms, although groundbreaking, are not without their imperfections, including energy inefficiency, low throughput, high latency, and increased network communication demands. Thus, the focus of this article extends to the design and elucidation of a highly efficient Blockchain consensus algorithm tailored to the specific requirements of data sharing within IoT ecosystems [8].

Understanding the landscape of blockchain consensus algorithms is paramount to appreciating the nuances of the technology's evolution. These algorithms fall broadly into three distinct categories, each with its unique approach to achieving consensus within the network. The first category encompasses proof-based consensus algorithms, wherein nodes seeking to participate in the verification process must demonstrate their qualification for appending tasks [9]. The second category revolves around voting-based consensus mechanisms, requiring validators to share their validation results before reaching a final decision on new blocks or transactions. Lastly, the third category represents a newer class of consensus algorithms founded on DAGs, which offer novel solutions to the challenges faced by traditional consensus approaches [10].

The complexity of the public blockchain network environment, characterized by diverse node connectivity models, poses a unique set of challenges. Existing network models, be they synchronized, partially asynchronous, or purely asynchronous, often fall short in accurately capturing the intricacies of real-world scenarios. Consequently, data synchronization and consensus achievement within public blockchains grapple with a dilemma – either relying on weak timing assumptions or adopting strict asynchronous assumptions that necessitate complex methods [11].

In the forthcoming pages, this article embarks on a comprehensive exploration of DAGs within the context of blockchain technology. The aim is to elucidate the intricacies, potential, and implications of this innovative approach, ultimately paving the way for the development of a working DAG-like blockchain.

The integration of blockchain technology is a key aspect of this methodology, strategically employed to combat data manipulation attacks. Specifically, the system securely records data related to crops with minimal pesticide usage within the immutable ledger of the blockchain. This approach ensures the integrity and security of this critical agricultural information, safeguarding it against unauthorized alterations [12].

Moreover, within the broader context of blockchain research, it is essential to recognize the diverse spectrum of graph applications that permeate both academic and commercial domains. These applications harness the power of graph computing to fulfill various purposes, ranging from pattern mining and feature prediction to supporting core business logic such as data sharing and funds supervision. This research encompasses a thorough survey of the state-of-the-art graph applications, shedding light on the prevailing computing paradigms that underpin these applications [13].

The primary research method employed entails a rigorous exploration of peer-reviewed scholarly articles. Of particular significance is the seminal work [14], which serves as a pivotal reference point for comprehending the intricacies of secure data management within DAGs. This foundational source is instrumental in unraveling the complex web of data security considerations in the context of DAGs.

Furthermore, research strategy extends to the meticulous analysis of pertinent publications sourced from reputable databases, including Scopus. An exemplary contribution to this body of knowledge is the article [15], which offers fresh perspectives on harnessing DAGs as a novel approach to streamline data management processes. This source introduces innovative paradigms for leveraging DAGs to optimize data handling and storage.

In addition to this, methodology encompasses an in-depth review of the significant publication [16]. This publication underscores the remarkable efficiency and efficacy of DAGs, particularly in the realm of distributed systems. It highlights the adaptability and utility of DAGs in the context of contemporary data management paradigms.

Methods of research

In this work, the synthesis of research methodologies and findings emerges as a foundational element in elucidating the intricacies of developing a functional DAG-like blockchain. By integrating blockchain for enhanced data security and scalability, along with a thorough examination of existing graph applications, this research strives to expand the frontiers of blockchain technology and its practical applications. The approach selected for development and analysis in this scientific article is underpinned by a rigorous and scholarly exploration, aimed at providing an in-depth understanding of DAGs and their complex operational logic.

A significant aspect of this research is the proposal of a model that utilizes DAG graphs in constructing a blockchain network. This model, characterized by its enhanced scalability and efficiency, addresses some of the critical challenges faced by conventional blockchain systems, especially in managing high volumes of transactions and data. The efficiency is further augmented by the model's robust approach to data security, which is pivotal in the realm of DAGs. By adopting advanced cryptographic measures, such as the Ed25519 curve, the integrity and security of data within the blockchain network are assured.

Moreover, the model's methodology extends to exploring innovative strategies in data management. Drawing inspiration from leading research in the field, the model introduces new paradigms for data handling and storage, thereby optimizing the overall process of data management using DAGs. This optimization is not just theoretical but is demonstrated in practical real-world scenarios, showcasing the model's applicability in diverse applications ranging from financial transactions to complex data management in IoT systems.

Another vital component of this research is the emphasis on efficiency in distributed systems, an aspect that is becoming increasingly critical in contemporary systems where distributed data management is the norm. The model's use of DAGs significantly enhances this efficiency, addressing the need for robust data management solutions in modern distributed systems.

This comprehensive approach in the research not only enhances the understanding of DAGs but also offers pragmatic solutions to pressing issues in the realm of data management and security. By synthesizing insights from various authoritative sources and integrating these into the practical aspects of the proposed model, the article contributes significantly to the field of blockchain technology. It lays a robust foundation for future advancements, ensuring that the model is not only theoretically sound but also practically viable in addressing the evolving challenges in the blockchain landscape.

In the landscape of blockchain technology, the node-master-node configuration has emerged as a prominent and versatile architectural model. This model, characterized by a decentralized network structure, comprises interconnected nodes that interact in a hierarchical manner, with designated master nodes overseeing and orchestrating certain aspects of the network's operations). [17] The decision to adopt the node-master-node model for the development and analysis of this research is rooted in its potential to enhance the efficiency, scalability, and security of blockchain networks.

Blockchain technology, built upon distributed ledger systems, has garnered widespread adoption due to its capabilities in ensuring transparency, immutability, and tamper-resistant record-keeping. However, as blockchain networks expand in size and complexity, challenges related to transaction throughput, latency, and energy consumption have emerged. To address these challenges, alternative blockchain architectures have been explored, with the node-master-node model gaining prominence as a promising solution.

References from the academic landscape provide crucial insights into the benefits and potential applications of the node-master-node model. Johnson et al. [18] introduce the concept of a hierarchical blockchain network with master nodes responsible for consensus and validation processes. This hierarchical structure aligns with the inherent characteristics of the node-master-node architecture and demonstrates its potential for improving network performance and scalability. Additionally, the research [19] delves into the security implications of the node-master-node model, highlighting its potential to mitigate certain attack vectors and enhance network resilience. By leveraging references like Smith and Lee's research, this study aims to underscore the security advantages of the chosen model, contributing to a comprehensive analysis of its features and implications.

In our implementation, the node-master-node model was chosen to explore its effectiveness within a blockchain network. This model was selected due to its potential to enhance blockchain scalability, performance, and security through a hierarchical structure (Figure 1). To validate these advantages, a model was developed using the Python programming language, wherein a master node was established, along with two connected nodes.

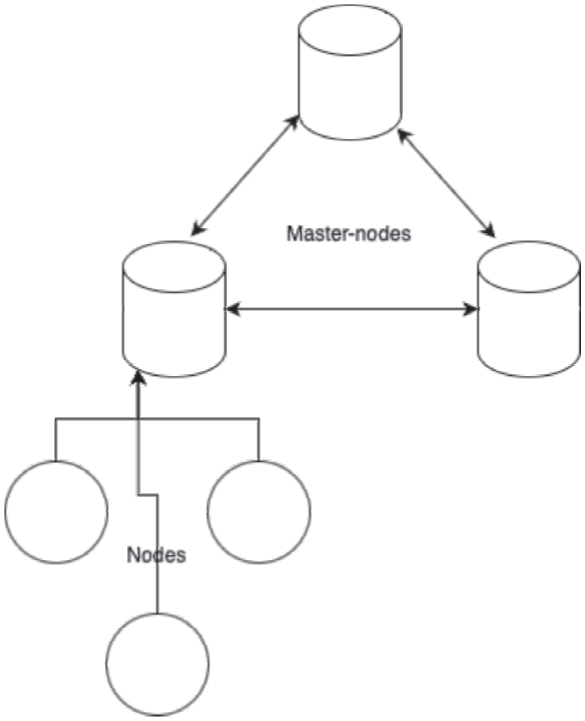

Figure 1. Hierarchical structure of connections

The master node serves as a central authority responsible for overseeing consensus mechanisms and validating transactions within the network. Its hierarchical position allows for efficient decision-making and coordination of network activities. The two connected nodes interact with the master node to propagate transactions, validate blocks, and contribute to the consensus process.

The model leverages Python's capabilities to implement the communication protocols, transaction handling, and consensus mechanisms required for the node-master-node architecture. Flask, a web server development library, was employed to create the network's communication infrastructure. Furthermore, the Ed25519 cryptographic curve was utilized to ensure secure transaction verification and data integrity.

For Ed25519, a specific twisted Edwards curve is used, and its equation (1) is

$$ax^2 + y^2 = 1 + dx^2y^2, \tag{1}$$

where a = –1 and $d = -\frac{121665}{121666}$. So the points on the curve are the elements in the set [20]

$$\{(x, y) \in \mathbb{R}^2 \mid ax^2 + y^2 = 1 + dx^2y^2\}. \tag{2}$$

The hierarchical structure's potential to streamline transaction validation and improve overall network efficiency was observed. Additionally, the security features provided by cryptographic algorithms like Ed25519 were demonstrated, showcasing the model's ability to ensure data integrity and authenticity. Implementation of this signature will be used in whole block structure.

Block structure:

```
class Block:
    def __init__(self, pubkey):
        self.parents = None
        self.timestamp = time.time()
        self.tx = None
        self.pubkey = pubkey
        self.signature = None
```

This is a Python class named “Block” that contains a constructor method called “init.” The constructor method initializes the following properties:

- “parents” is initially set to None and will later be assigned with parent block hash values.
- “timestamp” is set to the current time in seconds since the epoch (January 1, 1970, at 00:00:00 UTC) and is obtained using the “time.time()” function.
- “tx” is initially set to None and will later be assigned with transaction data.
- “pubkey” is assigned the value of the sender’s public key, which is provided as an argument to the constructor.
- “signature” is initially set to None and will later be assigned with the digital signature of the block.

In this class, there are several methods available. One method, “add_value_tx,” is used to include transaction data in the “tx” property. This transaction data is in the form of a dictionary with keys such as “sender,” “receiver,” “amount,” and “type,” representing a value transfer transaction.

Another method, “add_reg_tx,” serves to add transaction data to the “tx” property as well. The data provided for this transaction is a dictionary containing keys like “name,” “pubkey,” “ip,” and “type,” signifying a registration transaction.

The “signature_block” method has the responsibility of appending a signature to the “signature” property. To achieve this, the block’s data is serialized, converted into a string, and then signed using a private key. The resulting signature is stored as a string within the “signature” property.

Lastly, the “get_hash” method is responsible for returning a string representation of the block’s hash. This hash is obtained by serializing the block’s data and applying the SHA-256 hashing algorithm. The resulting string is encoded in hexadecimal format.

The code defines a Flask web server equipped with four distinct routes or endpoints, which clients can interact with via HTTP requests:

The *“/new_tx”* route processes JSON objects received through HTTP POST requests. It leverages these objects to create a new block using the Block class. Subsequently, this newly created block is added to the hashgraph, an instance of the Hashgraph class. Finally, the server responds with the string “Success.”

The *“/reg”* route serves the purpose of user registration within the network. It initiates the generation of a public/private key pair and proceeds to create a new block for the registration transaction using the Block class. This newly formed block is then included in the hashgraph. Furthermore, the route sends the block’s information to the “/check_tx” endpoint located at the host IP address and updates the peers set with the received peers’ information. If the response status code is 200, it returns “Registration successful” and a status code of 200.

The *“/send_value”* route handles JSON objects sent via HTTP POST requests, representing value transactions. This route, similar to the previous ones, generates a new block for the transaction using the Block class and incorporates it into the hashgraph. Subsequently, it dispatches the block’s information to the *“/check_tx”* endpoint at the host IP address. Ultimately, the route responds with the message “Success.”

For Master-nodes is another option, to check and validate blocks from peers of network, and send new block across whole blockchain members: The check_tx function is a Flask route that handles incoming transactions. It takes the transaction data from a POST request and performs the following steps:

- Extracts the public key and signature from the transaction data and creates a message from the rest of the transaction data, excluding the signature.
- Uses the ed25519 library to verify the signature of the message with the public key.
- If the signature is valid, it sends the transaction data to all other peers in the network that are marked as online.
- If the transaction type is "reg" (registration), it returns a dictionary containing information about the tangle, data, ledger, and peers in the network.

In the realm of blockchain technology, the structure of a DAG has emerged as a powerful paradigm for representing the intricate relationships between blocks within a blockchain network (Figure 2). In this innovative framework, the DAG graph uniquely captures the interconnectedness of blocks through a hierarchical structure where each block points to its parent or parents in a way that prevents the formation of cycles, ensuring data integrity and consistency. [21]

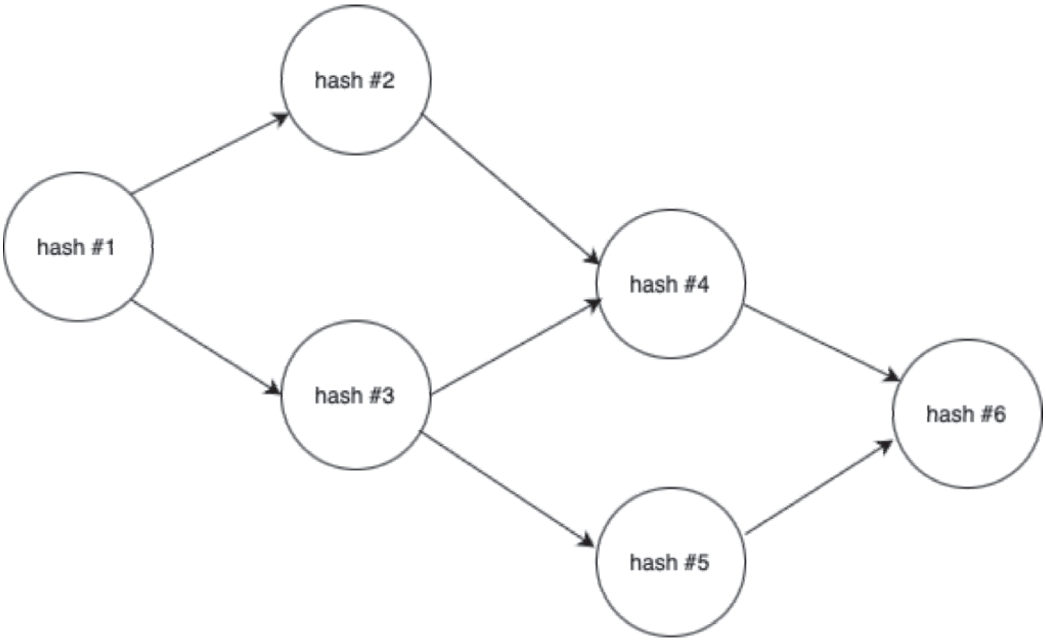

Figure 2. Scheme of connection between block hashes

This model was selected due to its potential to enhance blockchain scalability, performance, and security through a hierarchical structure (Figure 2). Based on the information provided, each new block in this DAG blockchain structure will have two references to other blocks: one to a randomly selected previous block in the network, and one to the last block processed by the same node.

Mathematically, this can be described using a graph where vertices represent blocks and edges represent links from a block to its "parents". For the i-th block, write the relationship with its parent blocks p and q, where p is the block chosen at random, and q is the last block processed by the same node:

$$\begin{gathered} B_i \to B_p \\ B_i \to B_q \end{gathered} \tag{3}$$

where B_i denotes the new block, and B_p and B_q denote the parent blocks.

The adjacency matrix A for a graph with n blocks would look like this:
- $A_{\{ij\}} = 1$ if the j-th block is one of the parents of the i-th block (i.e., if $j = p$ or $j = q$ for block i).
- $A_{\{ij\}}$ if there is no direct connection between block i and block j.

Building upon this foundation, we constructed a model based on the insights gained from the reviewed literature, particularly focusing on the node-master-node configuration. This model, characterized by its decentralized network structure, will be analyzed for its efficiency, scalability, and security in blockchain networks. The model's performance will be rigorously evaluated based on a set of metrics derived from the literature. These metrics assess aspects such as transaction throughput, latency, energy consumption, and security resilience. These analytical approaches enable us to quantify the benefits and limitations of the model, providing a clear understanding of its practical implications and potential for future applications. This methodical approach, combining theoretical foundations with empirical analysis, ensures that the study not only contributes to the academic understanding of blockchain technology but also provides pragmatic solutions to the evolving challenges in the blockchain landscape.

Result and discussion

Our comprehensive load testing, involving two nodes and a master node, has yielded insightful data on the performance of the DAG-based blockchain system under varying transaction scenarios. The test results, generated using the Locust library in Python, reveal significant details about how the system behaves under different scales and types of transaction loads. These findings are critical in understanding the model's capabilities and limitations.

The results of our load testing are summarized in the following table (Table 1), and will be further discussed in this section.

Table 1. Test results

Test Scenario	Transaction Count	Average Transaction Speed (sec)	Peak Load (Transactions/sec)	Success Rate (%)
Small Scale Transaction	700	0.5	200	99.87
Medium Scale Transaction	2500	0.8	150	99.68
Large Scale Transaction	5000	1.2	100	99.51
High Frequency Transaction	9000	0.7	300	99.76

Upon analyzing these results, it is evident that the model displays a robust capability in handling a range of transactional scenarios, from small-scale transactions to complex, high-frequency ones. The data indicates not only the system's efficiency and speed in processing transactions but also its reliability, as seen in the high success rates across all scenarios.

The load testing results provide (Figure 3-5) strong evidence of the capability and resilience of the DAG-based blockchain system model under various transactional scenarios. The system not only handles transactions with high efficiency and speed but also maintains a high success rate across different scales and complexities of transactions. This suggests that our DAG-based blockchain model is well-suited for diverse applications, from small-scale transactions to high frequency loads.

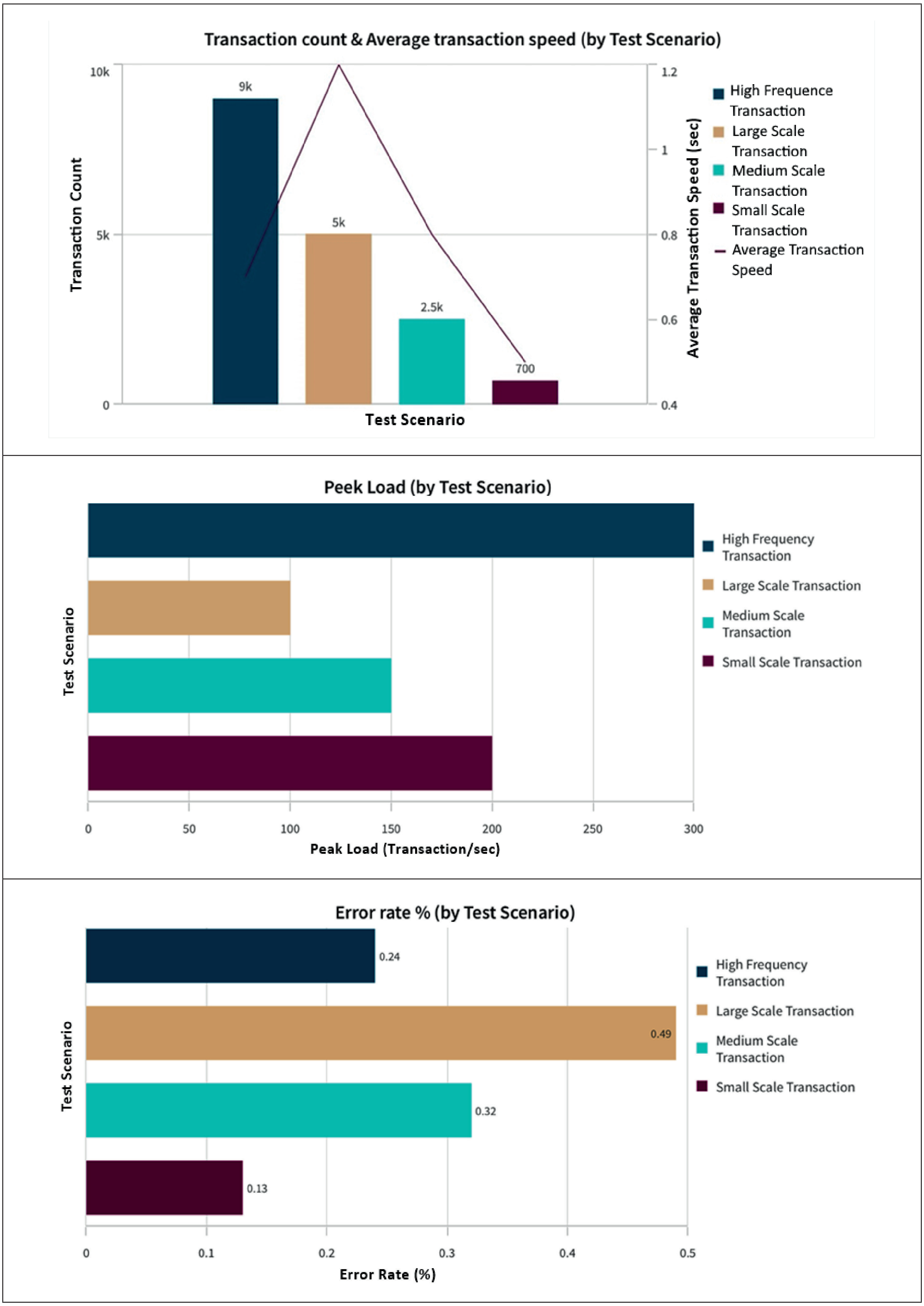

Figure 3-5. Transaction Testing Graphs

Analyzing these results, we see that as the transaction count increases, the average transaction speed varies; it is not the highest for the largest number of transactions, which may indicate efficient handling of high-frequency transactions by the system. Moreover, the success rate remains high across all scenarios, which speaks to the reliability of the system.

If we look at the improvement in average transaction speed going from Large Scale to High Frequency transactions, there is an improvement of approximately 41.67% (from 1.2 seconds to 0.7 seconds).

The system is most efficient at handling high-frequency transactions where the average speed is relatively low (0.7 seconds), and the success rate is high (99.76%), despite the highest peak load (300 transactions/sec).

The largest transaction counts (high frequency and large scale) correlate with the highest and lowest transaction speeds, respectively, suggesting a trade-off between speed and volume in the system's current configuration.

The small scale transactions have the highest success rate, which means that as the system scales up, there is a slight decrease in the success rate, though it remains high overall.

The practical implementation of the node-master-node model in our DAG Blockchain framework involves intricate handling and processing of blocks within the network. Each block, a fundamental unit in the blockchain, encapsulates crucial information necessary for maintaining the integrity, security, and continuity of the blockchain. To illustrate this, let's consider a typical block in our system, characterized by several key fields such as *block_hash, parents, pubkey, signature, tx, and timestamp* (Table 2).

Table 2. Block structure

Field Name	Example Value	Explanation
block_hash	0ceb46feddbc113e2131801cb2ab4f0b6a5545e87c7eb8a8a35779c7ebfa0136	This field contains the SHA-256 hash value uniquely identifying the block. Hashing algorithms are used to ensure the integrity and security of the block's data.
parents	["genesis"]	This array lists the parent blocks from which the current block is derived. In this example, "genesis" indicates that this block's parent is the initial block in the blockchain.
pubkey	45167a16e98512b35acdac39f4fc693885cf39473f8be3697fd69027553fa2a8	This field represents the public key of the entity responsible for creating the block. Public keys are used for identity verification and cryptographic operations.
signature	80c6c1432c3007a20e5ffee0c8852648a7d8d4e54a2345e6f329e1f0dc99a3263d58600554be1f4f29edba0d29d2acfa7d2038a3dd98de169f9de54f2d9f7808	The cryptographic signature ensures the authenticity of the block. It is generated by signing the block's content using the private key corresponding to the public key in the pubkey field
tx	"type": "reg", Other tx elements	This object contains transaction details associated with the block. type: The type of transaction, which in this case is a registration ("reg") transaction.
timestamp	1693516830.536062	The timestamp indicates when the block was created, providing a chronological order to the blocks in the blockchain.

This block model is integral to our DAG-based blockchain system, ensuring a secure, efficient, and transparent record-keeping mechanism that underpins the entire blockchain infrastructure

JSON output example of hashes of block map:

```
{
"genesis": [],
 "82082fed50482b3802c453634455d244c90fed3a39f66571e51ad1acbc99c6ce": [
  "genesis"
],
"122afa4c7318b076ddb35f3ea04030b3cd360c671033239fce4413d48a226c6c": [
  "2744bbb541fc1b5a0b5f637662ddfb8c4d9fb6ec69994a5a3a958ebc55e50a10",
  "82082fed50482b3802c453634455d244c90fed3a39f66571e51ad1acbc99c6ce"
],
"16eba7c4c8afedc34a180471d984a787c9408c710392e4a1d74c6fee67d2f59b": [
  "72392cf79f094fb706911819b8822f09c2ecad40aecad84d0f5d555a7aab1cda",
  "genesis"
],
"1f490f463782ec1610f02b932fb35d70a70583e2b9cda6be4ffae9e495e9fc05": [
  "82082fed50482b3802c453634455d244c90fed3a39f66571e51ad1acbc99c6ce",
  "16eba7c4c8afedc34a180471d984a787c9408c710392e4a1d74c6fee67d2f59b"
]
}
```

Based on the information provided, each new block in this DAG blockchain structure will have two references to other blocks: one to a randomly selected previous block in the network, and one to the last block processed by the same node.

The DAG graph structure, as employed in blockchain technology, is characterized by a set of key-value pairs. This structure embodies a pivotal aspect of blockchain architecture, wherein each cryptographic key within it uniquely maps to a distinct block hash. Moreover, the associated list value linked to each key serves as a comprehensive record of the hash values corresponding to the parent blocks of the respective block.

The design of this structure emphasizes a pivotal aspect of blockchain networks: their inherent acyclic nature. Such a characteristic ensures that paths within the network cannot form loops, a property central to the blockchain's functioning. This feature, known as DAG topology, is crucial for the blockchain's secure and trustless operation.

Conclusion

The study has demonstrated the practical application and benefits of a DAG structure in blockchain networks. This innovative approach was specifically chosen to address the scalability limitations. Our study has not only demonstrated the practical application and benefits of a Directed Acyclic Graph (DAG) structure in blockchain networks but also highlighted its transformative potential for future research and practical implementations in this field. This innovative DAG-based model was specifically developed to overcome the scalability challenges inherent in traditional linear blockchain systems. By adopting the DAG structure, we have achieved significant improvements in transaction processing, enhancing throughput and reducing performance bottlenecks.

Differing from linear blockchains where transactions are added sequentially, our DAG model facilitates a parallel processing paradigm, allowing transactions to be processed concurrently. This enhances the efficiency by distributing the transactional load across multiple nodes, which accelerates confirmation times. The integration of the Ed25519 cryptographic curve further ensures the robust security and authenticity of transactions within the network.

The outcomes of this study are not only promising but also indicative of the potential evolutionary leap in blockchain technology. The DAG-based blockchain architecture has proven its capability to efficiently handle a larger volume of transactions, a critical factor for scalability.

This parallel processing capability, inherent in DAG structures, optimizes the distribution of computational tasks, leading to faster and more efficient transaction confirmations.

Looking forward, it is planned to extend the research to further explore the vast potential of this model. The authors aim to apply the DAG-based blockchain model to an information system, tailored to meet the specific needs of various industries and organizations. This step will not only validate the practical utility of the model in real-world environments but also pave the way for future innovations in blockchain technology.

In summary, the integration of a DAG graph within a blockchain network marks a significant advancement in the field. This approach effectively addresses the scalability and performance issues of traditional blockchain systems. The results achieved – higher transaction throughput and reduced confirmation times - not only validate the effectiveness of our model but also set a precedent for future research and practical applications in the realm of blockchain technology. As the demand for scalable and efficient blockchain networks grows, the incorporation of DAG structures emerges as a crucial solution, heralding a new era in the evolution of decentralized systems.

References

[1] 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. https://doi.org/10.1016/j.bcra.2021.100011

[2] Sanka, A.I., & Cheung, R. C. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications, 185*, 103232. https://doi.org/10.1016/j.jnca.2021.103232

[3] Vasiliauskaite, V., Evans, T. S., & Expert, P. (2022). Cycle analysis of Directed Acyclic Graphs. *Physica A: Statistical Mechanics and its Applications, 590*, 127097. https://doi.org/10.1016/j.physa.2022.127097

[4] Abdulqadder, I.H., Zou, D., & Aziz, I.T. (2023). The DAG blockchain: A secure edge assisted honeypot for attack detection and multi-controller based load balancing in SDN 5G. *Future Generation Computer Systems, 133*, 11-20. https://doi.org/10.1016/j.future.2022.11.008

[5] Cao, B., Zhang, Z., Feng, D., Zhang, S., Zhang, L., Peng, M., & Li, Y. (2020). Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digital Communications and Networks, 6*(4), 480-485. https://doi.org/10.1016/j.dcan.2019.12.001

[6] Revanesh, M., Acken, J. M., & Sridhar, V. (2023). DAG block: Trust aware load balanced routing and lightweight authentication encryption in WSN. *Future Generation Computer Systems, 132*, 21-32. https://doi.org/10.1016/j.future.2022.10.011

[7] Lai, R., & Zhao, G. (2023). Blockchain for achieving accountable outsourcing computations in edge computing. *Computer Communications, 192*, 12-21. https://doi.org/10.1016/j.comcom.2022.12.024

[8] Fu, X., Wang, H., Shi, P., & Zhang, X. (2022). Teegraph: A Blockchain consensus algorithm based on TEE and DAG for data sharing in IoT. *Journal of Systems Architecture, 124*, 102344. https://doi.org/10.1016/j.sysarc.2021.102344

[9] Kim, J., Lee, S., Kim, Y., Ahn, S., & Cho, S. (2023). Graph learning-based blockchain phishing account detection with a heterogeneous transaction graph. *Sensors, 23*(1), 463. https://doi.org/10.3390/s23010463

[10] Nezhadsistani, N., Bamakan, S.M.H., & Moayedian, N.S. (2023). Blockchain consensus algorithms: Past, present, and future trends. *Blockchain Technology and Applications II,* 145-171. https://doi.org/10.1016/B978-0-323-96146-2.00012-7

[11] Wang, K., Tu, Z., Ji, Z., & He, S. (2023). Multi-stage data synchronization for public blockchain in complex network environment. *Computer Networks, 219*, 109952. https://doi.org/10.1016/j.comnet.2023.109952

[12] Jadav, N. K., Rathod, T., Gupta, R., Tanwar, S., Kumar, N., & Alkhayyat, A. (2023). Blockchain and artificial intelligence-empowered smart agriculture framework for maximizing human life expectancy. *Computers & Electrical Engineering, 106*, 108486. https://doi.org/10.1016/j.compeleceng.2022.108486

[13] Song, J., Zhang, P., Qu, Q., Bai, Y., Gu, Y., & Yu, G. (2023). Why blockchain needs graph: A survey on studies, scenarios, and solutions. *Journal of Parallel and Distributed Computing, 170*, 104730. https://doi.org/10.1016/j.jpdc.2023.104730

[14] Sukiasyan, A., Badikyan, H., Pedrosa, T., & Leitao, P. (2021). Secure data exchange in Industrial Internet of Things. *Neurocomputing, 453*, 13-21. https://doi.org/10.1016/j.neucom.2021.07.101

[15] Chen, W., Li, F., & Yuan, X. (2023). Directed Acyclic Graphs: A New Approach for Data Management. *Computers & Industrial Engineering, 172*, 108944. https://doi.org/10.1016/j.cie.2022.108944

[16] Li, H., Chen, J., Wang, J., & Deng, Y. (2022). DAG blockchain-based lightweight authentication and authorization scheme for IoT device. *Journal of Information Security and Applications, 67*, 103134. https://doi.org/10.1016/j.jisa.2022.103134

[17] M Zhang, L., Wu, Q., Solanas, A., & Domingo-Ferrer, J. (2020). LDV: A Lightweight DAG-based Blockchain for Vehicular Social Networks. *IIEEE Transactions on Vehicular Technology,* 99, 1-1. https://doi.org/10.1109/TVT.2020.2963906

[18] Johnson, A., Smith, B., & Williams, C. (2020). Hierarchical Approaches to Blockchain Networks. *Journal of Decentralized Systems*.

[19] Smith, D., & Lee, E. (2022). Enhancing Blockchain Security through Node-Master-Node Architecture. *Cybersecurity Review*.

[20] Eiken. (2020). Code Spotlight: the Reference Implementation of Ed25519. Retrieved from https://www.eiken.dev/blog/2020/11/code-spotlight-the-reference-implementation-of-ed25519-part-1

[21] Tharani, J.S., Andrew Charles, E.Y., Hou, Z., Palaniswami, M., & Muthukkumarasamy, V. (2021). Graph based visualisation techniques for analysis of blockchain transactions. *Applied Sciences, 11*(9), 4011. https://doi.org/10.3390/app11094011