Evaluating Blockchain Consensus Mechanisms: A Comprehensive Three-Tiered Approach with Application to IOTA 2.0

***Abstract*—We introduce a three-tiered approach to validate blockchain consensus mechanisms, exemplified through the IOTA 2.0 protocol. Traditional methods often struggle to link theoretical constructs with real-world performance, a gap our methodology aims to bridge. We offer a comprehensive framework that transitions from abstract concepts to empirical insights by integrating theoretical models, simulation environments, and real-node testing. Applying this to IOTA 2.0, we validate its consensus mechanism across theoretical, simulated, and practical scenarios, demonstrating our method's ability to enhance the understanding and optimization of distributed ledger technologies (DLTs). This work provides a scalable and efficient framework for future DLT evaluations, contributing significantly to the field's advancement.**

***Index Terms*—simulation, blockchain, consensus protocol, distributed ledger technology, performance analysis, directed acyclic graph**

I. Introduction

In blockchain technology's rapidly evolving landscape, the significance of simulation and modeling has become increasingly paramount. Since the inception of Bitcoin [1], blockchain technology has attracted significant attention, proposing revolutionary possibilities for financial systems and other sectors needing secure, decentralized solutions. Despite the high expectations, the real-world impact of blockchain on traditional economic sectors has been unfolding more gradually, with adoption and integration facing various hurdles. This reality makes the performance evaluation of these large-scale distributed systems challenging and costly. It highlights the critical importance of blockchain simulators, which serve as indispensable tools in replicating complex real-world processes at significantly lower costs and with greater efficiency [2].

A. Related Work

SimBlock [3], an open-source blockchain network simulator, addresses the challenges of conducting experiments with actual blockchains that necessitate a vast network of nodes spread across extensive areas. Its effectiveness is demonstrated by validating its simulations against actual blockchain data and showcasing practical applications, such as analyzing the impact of neighbor node selection algorithms and relay networks on block propagation time.

To address the existing limitations of SimBlock, particularly its inability to simulate block mining and incentive mechanisms, an improved version of the simulator was proposed in [4]. This enhanced version effectively abstracts the current Bitcoin blockchain, significantly broadening its utility across various domains, including smart cities. Experimental analysis confirms the effectiveness of these improvements and indicates that integrating relay network modeling could further refine the accuracy of SimBlock's simulations. Building upon this, [5] explores the development and application of SimBlock in large-scale blockchain research, focusing on studies on blockchain performance and security.

In [6], a novel quantitative evaluation method is introduced for blockchain protocols, offering an objective analysis amidst the diversity in protocol characteristics and performance. It is crucial to acknowledge the challenges in developing a universal simulator that can accommodate a variety of applications, purposes, and underlying data structures. For Directed Acyclic Graph (DAG)-based blockchain protocols, specialized simulators such as [7]–[10] have been crafted to analyze the performance and security of the IOTA's Tangle [11].

B. Our Contributions

In this paper, we introduce a novel three-tiered methodology that bridges the gap between theoretical mathematical models, high-level simulations, and practical insights from real-world testing. This methodology addresses the challenge of transforming complex theoretical frameworks into empirically verifiable insights through practical experimentation. It integrates three distinct yet complementary tiers: (i) foundational theoretical models, (ii) sophisticated simulation environments, and (iii) direct real-node testing, benefiting end-users.

To demonstrate the effectiveness of our methodology, we apply it to the IOTA 2.0 consensus mechanism. At the mathematical model tier, we define theoretical boundaries for various scenarios. At the simulation tier, we implement core functions to estimate results within complex environmental setups, considering parameters like node count, network latency, and attacker behavior. Building on previous works that established the first two tiers, the mathematical model [12] and high-level simulator [10], our contribution adds a crucial third tier: real-world log analysis and profiling. We collect logs from 64 VMs in a cloud environment, each running IOTA-core software[1] on a 4-core AMD EPYC Processor with 8GB of RAM, to test under various conditions and monitor performance metrics like CPU/memory usage, throughput, and block confirmation times. In this paper, we concentrate on

[1]IOTA-core, available at https://github.com/iotaledger/iota-core, 2024.

elucidating the methodology rather than comparing testing outcomes from the node software with those from other tiers. This approach is necessitated by the fact that the IOTA-core, which serves as the node software, is currently in an active development phase. It is subject to ongoing fine-tuning, substantial bug fixing, and enhancement efforts to achieve optimal performance and reliability.

Our methodology's applicability extends beyond the IOTA consensus mechanism. By abstracting DAG-based consensus principles into a versatile framework, we can evaluate the performance and resilience of various DAG-based networks. This modular approach promotes a unified and thorough evaluation process adaptable to diverse blockchain technologies. Employing our comprehensive methodology, we conduct a comparative analysis of actual IOTA-core network logs against earlier simulation results, validating the Tangle 2.0 protocol and its Delegated Proof-of-Stake validator mechanism and demonstrating our approach's practical viability.

C. Outline

This paper is organized as follows: Section II outlines the proposed three-tiered analysis methodology. Section III delves into the validator mechanism and the Tangle 2.0 consensus protocol, which are used as case studies to demonstrate our methodology. Section IV utilizes the confirmation time of IOTA 2.0 as a case study to illustrate our methodology. Finally, Section V discusses the broader implications of our findings and concludes the paper.

II. Three-Tiered Analysis Methodology

Our research introduces an integrative three-level modeling framework specifically crafted to rigorously test the performance and robustness of DAG-based Distributed Ledger Technologies (DLTs). This framework is designed for evaluating and enhancing the efficiency, security, and resilience of consensus mechanisms across a broad spectrum of DAG-DLT systems. We aim to advance the DLT field by ensuring these technologies can meet the challenges of security and operational demands through comprehensive testing and analysis across various platforms and systems. This proposed multi-tier analysis methodology mirrors the multi-tiered approaches found in other fields, such as layered modeling in vehicle design, layered methodology in medical research [13], semiconductor chip design [14], etc. It underscores the novelty and necessity of applying such multidimensional modeling techniques in the crypto domain, a practice not yet conventional but incredibly potent in its potential to transform our understanding and optimization of DLTs. In this section, IOTA serves as a case study to demonstrate our methodology. It was selected for its well-documented presence across three distinct tiers that are both existing and public: the mathematical model as outlined by Müller et al. [12], the high-level simulator TangleSim [10], and the practical real-node implementation, IOTA-core[1].

A. Mathematical Model Tier

The mathematical model lies at the foundation of our framework, a construct designed to abstract the complexities of the Tangle consensus into a quantifiable and analyzable form. This model encapsulates various parameters, including network latency, throughput, and node behavior, thereby providing a solid baseline for theoretical exploration. One of its key strengths is the ability to distill the essence of complex interactions into a form suitable for rigorous analysis. Müller et al. [12] provides a notable example of this, analyzing liveness and safety under various theoretical communication and adversary models, along with formal proofs. However, to delve into the consensus mechanism with finer granularity, assess diverse scenarios with customized parameters, and obtain practical simulation results based on specific assumptions, the implementation of a high-level simulation model becomes essential.

B. High-Level Simulation Tier: TangleSim

Ascending from theory to practice, the high-level simulator TangleSim serves as a vital intermediary. This simulator translates the theoretical parameters and concepts of the mathematical model into a simulated environment. TangleSim [10] enables the replication of the Tangle network within a controlled setting, thereby facilitating an in-depth analysis of consensus dynamics, network behavior, and data interactions. In TangleSim [10], only the core components of the consensus mechanism are implemented. This approach allows for decoupling from the current real node, which is subject to frequent updates, and enables the generation of more accurate and stable preliminary simulation results. These results not only validate the mathematical model but also provide crucial insights for the real-node implementation. TangleSim augments our understanding of the Tangle by integrating practical elements like network conditions and node dynamics, offering a more nuanced perspective than purely theoretical models. The primary aim of this high-level simulator is not to produce exact results but to provide accurate, stable preliminary simulations and insights for real node implementation.

C. Practical Implementation Tier: IOTA-core

The final tier of our framework is realized through IOTA-core[1], which moves beyond a model to become the actual implementation of the Tangle node used by end-users. IOTA-core serves as a critical link to real-world applications, enabling us to glean vital insights by analyzing operational data and logs from a functioning network. This tier is especially important as it is where our theoretical models and high-level simulations are tested in real-world scenarios. Engaging with a live network setting, IOTA-core sheds light on important issues such as performance limitations, design flaws, and other practical challenges. It transforms our theoretical and simulated plans into a concrete, working system. By delivering tangible results from extensive testing, IOTA-core establishes itself as the definitive tool for validating the reliability and efficiency of the Tangle consensus mechanism.

D. Bridging Tiers: Systematic Parameter Translation

Our three-tiered methodology is meticulously designed to facilitate a seamless translation and alignment of parameters across distinct levels of analysis, encapsulating the theoretical, simulation-based, and implementation phases. At the core of this bridging process lies a sophisticated mathematical model delineating theoretical boundaries, serving as the foundational tier. This model is instrumental in identifying key parameters and their theoretical limits, setting the stage for subsequent tiers. Following this, the TangleSim, positioned as the second tier, offers a more granular observation of these parameters in simulated environments. By replicating real-case scenarios focusing on core functionalities, TangleSim enables the refinement of our theoretical assumptions, allowing for a more precise estimation of system performance and robustness. This simulation tier effectively narrows the scope to essential parameters, facilitating a targeted and efficient analysis.

Transitioning to the third tier, the IOTA-core implementation embodies the culmination of our methodology, where the theoretical insights and simulated observations are applied to real-world configurations. This tier validates our parameter choices and theoretical models against tangible outcomes, ensuring that the final system configuration not only adheres to our rigorous standards but also demonstrates the anticipated levels of performance and robustness in practical deployments. The systematic parameter translation across these tiers is characterized by a cyclic enhancement process, where insights from the IOTA-core implementation feedback into the theoretical modeling foster a dynamic, iterative refinement of our analysis framework. This holistic approach ensures a disciplined and structured progression, mirroring methodologies employed in advanced technological and design fields, thereby guaranteeing the integrity and coherence of our consensus analysis across all levels of investigation.

E. Comprehensive Analysis: Performance, Security, and Robustness

Our methodology encompasses a detailed analysis of the Tangle's performance, security, and robustness. Utilizing TangleSim and the data and logs gathered from IOTA-core, we evaluate the protocol design from various angles. This extensive analysis is instrumental in establishing performance targets and hardware specifications, confirming the consensus mechanism's suitability for real-world deployment. Our framework encourages a dynamic, iterative development process, reflecting the collaborative and interdisciplinary nature of contemporary technology advancement.

III. Consensus in IOTA 2.0

The IOTA 2.0 consensus mechanism uses a DAG structure, named the *Tangle*, to ensure reliable broadcasting of blocks.

In a blockchain, each block contains a record of multiple transactions. Each block is securely linked to the previous block in the chain using a cryptographic hash function. This creates a chain of blocks that cannot be changed without changing all subsequent blocks in the chain, which requires the consensus of the entire network. As new transactions are added to the network, the chain of blocks grows, resulting in an immutable record of all transactions on the blockchain.

A. The Tangle

In a DAG-based DLT, blocks can be linked to more than one previous block, creating a DAG of blocks instead of a chain. The data structure consists of blocks and references from more recent blocks, called *children*, to older blocks, called *parents*. Blocks not referenced yet are known as *tips*. The Tangle starts with a genesis block, and nodes attach new blocks to previous blocks using an algorithm called *tip selection*.

As in a blockchain, these references to previous parts of the DAG allow for defining a certain confidence level for including the blocks in the final data structure. In the IOTA 2.0 consensus protocol, [12], the level of endorsement of blocks and their associated transactions are defined by the *Approval Weight*.

B. Validators and validation blocks

In the implementation of the IOTA 2.0 proposal, we distinguish between validators and block issuers, where validators regularly issue *validation blocks* that serve for validation only. Tip selection algorithms for basic blocks and validation blocks are different: basic blocks uniformly randomly choose other blocks to reference among all tips; in addition to those references, validation blocks uniformly randomly choose tips that indirectly reference the last validation block to ensure that validation blocks are well connected.

C. Confirmation

The Approval Weight of a block is calculated as the number of all validators referencing (directly or indirectly) the given block. A block is said to be *pre-confirmed* if referenced by a supermajority of all validators, i.e., more than $2/3$ of the number of validators. A block is *confirmed* when it receives approval from a total supermajority of pre-confirmed validation blocks. The transition from pre-confirmed to confirmed represents a further step in the consensus process. This stage is crucial for a block's finality within the network. Previous work did not address a finality notion, so the notations in [10] and [12] called pre-confirmed block confirmed blocks. Let us note that this covers only the aspects of confirmation of blocks; the mechanism of confirmation transactions is more involved as they use an active voting procedure to ensure conflict resolution, see [12].

IV. Analyzing Confirmation Times: A Case Study

As previously mentioned, the real node implementation for our case study, focusing on IOTA 2.0, is in a phase of rapid development and bug resolution. Nonetheless, our primary aim is to elaborate on our methodologies and processes for analyzing confirmation times across three distinct tiers. In this paper, we take the confirmation time of IOTA 2.0 as an illustrative example.

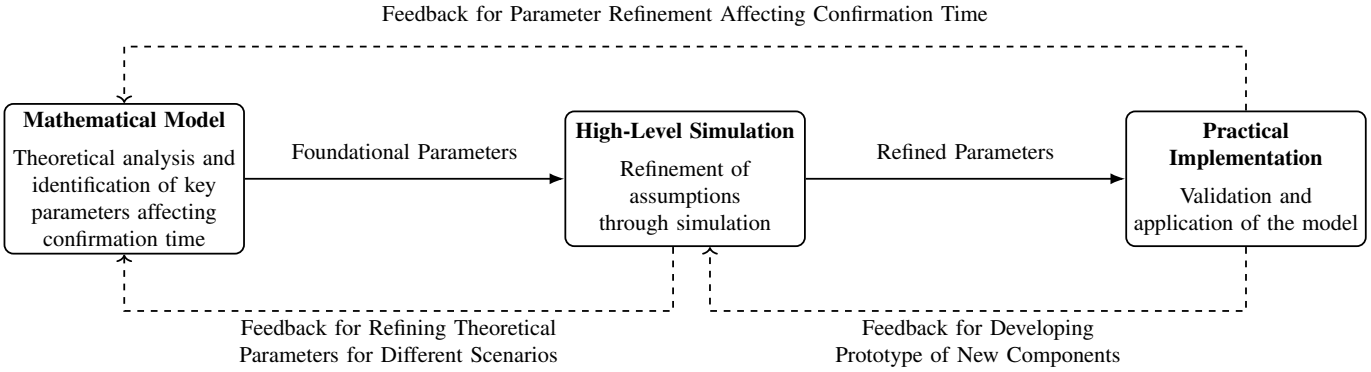

Fig. 1. Three-Tier Integration Approach for Analyzing Confirmation Times.

A. Analysis Flow

To analyze confirmation time, the flow chart depicted in Figure 1 illustrates a comprehensive three-tiered integration process designed to validate and refine theoretical models through high-level simulation and practical implementation. This iterative and dynamic approach ensures that the development and analysis are grounded in both theoretical foundations and practical validations, facilitating a robust understanding and optimization of the system under study.

At the foundation of this process lies the Mathematical Model tier, where theoretical analysis and key parameter identification take place. This initial stage is crucial for establishing a theoretical framework that guides the entire study. It involves a detailed exploration of the system's fundamental principles and the identification of key parameters that influence its behavior. This theoretical groundwork not only sets the stage for further investigation but also outlines the boundaries within which the system operates. We will use the theoretical upper bounds for the expectation of pre-confirmation time τ_{pre} developed in [12, Example 5]:

$$\tau_{\mathrm{pre}} \lesssim \frac{1}{\log k} h \log(\lambda h) + \frac{N}{\lambda}\left(1 - \log(1-\theta)\right); \qquad (1)$$

where $k = 64$ is the number of references of validation blocks, h is the network latency, $\lambda = 32$ the number of validation blocks per seconds, $N = 32$ the number of validators, and $\theta = 2/3$ the confirmation threshold. For $h = 100$ms, this yields an upper bound for the average pre-confirmation time of 2.14s.

Following the theoretical modeling, the process advances to the High-Level Simulation tier. Here, the assumptions and parameters derived from the mathematical model are refined through simulated environments. This stage allows for exploring the model's behavior under various conditions, providing insights that are not readily apparent from theoretical analysis alone. The high-level simulation serves as a bridge between theory and practice, enabling researchers to test hypotheses, explore the impact of different parameters, and refine their understanding of the system in a controlled yet flexible setting.

The third and final tier is the Practical Implementation, where the theoretical models and simulation findings are validated and applied in a real-world context. This stage is critical for testing the feasibility and effectiveness of the theoretical insights and simulation outcomes. The practical implementation provides concrete evidence of how the system functions in reality, allowing for the validation of the theoretical model and the refinement of simulation assumptions based on real-world data and feedback.

Integral to this three-tiered process are the feedback loops that connect each tier, promoting a continuous cycle of refinement and improvement. The first feedback loop, from Practical Implementation to Mathematical Model, enables the refinement of the theoretical model based on practical findings. This loop ensures that theoretical assumptions consistently align with real-world behaviors, enhancing the model's relevance and accuracy.

The second feedback loop, from Practical Implementation to High-Level Simulation, focuses on developing prototypes for new components. This feedback allows the simulation to incorporate practical insights and innovations, fostering a more accurate and comprehensive simulation environment.

Lastly, the feedback from High-Level Simulation to Mathematical Model, dedicated to refining theoretical parameters for different scenarios, ensures that the theoretical model remains dynamically aligned with the insights gained from simulation studies. This loop allows for the continuous adaptation and refinement of the model, ensuring its applicability across various scenarios and enhancing its predictive accuracy.

B. Parameter Refinement

Within the mathematical model, we explored a variety of parameters, such as network delay, number of tips, number of nodes, block issuance rate, and estimated time to confirmation, across different scenarios. These discussions help to set crucial theoretical limits for each case. TangleSim incorporates these vital parameters to implement the core components of IOTA 2.0, thereby allowing the observation of behaviors with greater granularity. The simulator enables us to identify the theoretical bounds of confirmation times and to precisely analyze the actual confirmation times influenced by variables like the number of nodes, network delays, number of neighbors, and

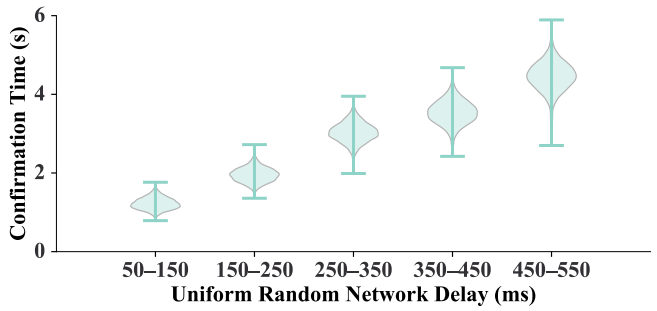

Fig. 2. Confirmation time distributions with different uniform random network delays [10]. Note that the confirmation time in the figure corresponds to the pre-configuration time in this paper.

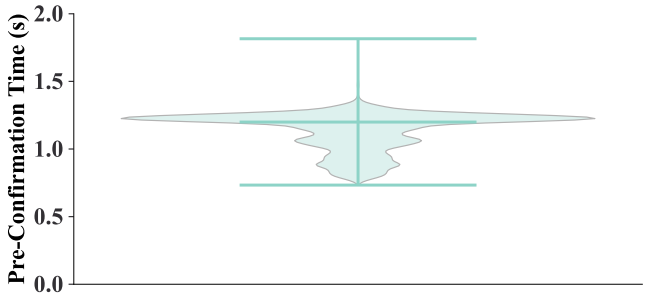

Fig. 3. Pre-Confirmation time distribution in the real node implementation.

topology. Furthermore, it facilitates the detailed visualization of the confirmation time distribution.

For instance, Fig. 2 illustrates the distributions of confirmation times across various network delays. The simulation setup for the figure is shown in Table I, as explained in [10].

TABLE I
SETTINGS OF PARAMETERS FOR THE SIMULATION

Item	Value
Validator node count	100
Non-validator node count	0
Number of neighbors	8
Validation blocks per second	100
Package loss	0
Network Latency (ms)	100
Maximal parent count	8

It should be noted that Fig. 2, which is based on results from the TangleSim study, actually depicts the pre-confirmation time as defined by our updated consensus mechanism, as detailed in the preceding section. This distinction is crucial for understanding the metrics used in our analysis and their implications for system performance.

Table II outlines the configurations used for the real node implementation in the experiment. It specifies the setup parameters, including the counts of validator and non-validator nodes, the number of neighbors each node is connected to, the rate at which validation blocks are processed (in blocks per second), package loss rate, network delay, and the maximum parent count for validation and non-validation blocks.

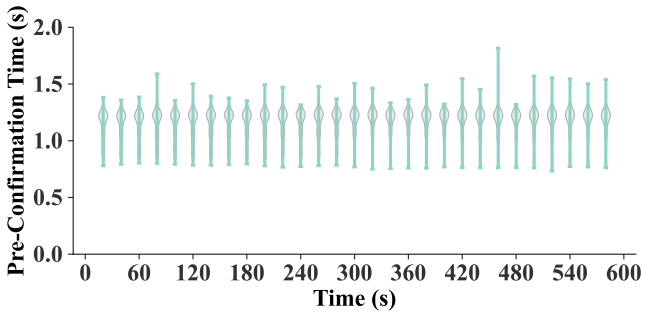

Fig. 4. Pre-confirmation time distributions of every 20 seconds in the real node implementation.

TABLE II
SETTINGS OF PARAMETERS FOR THE REAL NODE IMPLEMENTATION

Item	Value
Validator node count	32
Non-validator node count	32
Number of neighbors	6
Validation blocks per second	32
Base blocks per second	20
Package loss	0
Network Latency (ms)	100
Maximum validation block parent count	64
Maximum basic block parent count	8

The comparison between simulated and real-world data can reveal significant insights. The high-level simulator can approximate the results of the confirmation time distribution across different node counts, network latencies, etc. It can offer more direct suggestions for real node implementation adjustments when the test results are not satisfying.

Note that for node counts of 100 or less, the distribution of pre-confirmation times generated by TangleSim remains consistent; therefore, we have chosen not to repeat these results in our presentation. This consistency underscores the scalability of the consensus mechanism within smaller network sizes.

Figures 3 and 4 illustrate the pre-confirmation times collected from the logs of real node implementation. These figures reveal that the distribution of pre-confirmation times closely mirrors the distribution observed in Figure 2, highlighting a consistent pattern across both simulated and real-world scenarios. The pre-confirmation times in both methods stay under the theoretical upper bound in (1). This consistency underscores the reliability of the simulation model in predicting real node behavior, offering valuable insights into the effectiveness of the consensus mechanism under study.

Despite the ongoing development of the practical implementation and the absence of conclusive results with diverse scenarios, our methodology effectively uncovers critical bugs

and inefficiencies in the real node implementation[2]. This process is essential for guiding future optimizations and bug-fixing efforts.

Recognizing and addressing these variances is crucial, serving academic interests and practical needs in refining the real node implementation. By accurately identifying when and why these discrepancies arise, we can pinpoint potential areas for improvement. For example, longer than expected confirmation times may highlight problems with block propagation or validation algorithms, necessitating further investigation and adjustments.

V. Conclusion

This paper introduces a three-tiered methodology for analyzing and improving consensus mechanisms in DAG-based Distributed Ledger Technologies (DLTs). Through a case study on confirmation time analysis in IOTA 2.0, we demonstrate the efficacy of our approach and its potential to uncover vital insights for the ongoing development and enhancement of DLT systems. This method significantly advances our understanding of consensus mechanisms in practical scenarios and creates a versatile framework for assessing diverse DLTs.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." 2008.

[2] R. Paulavičius, S. Grigaitis, and E. Filatovas, "An overview and current status of blockchain simulators," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1–3.

[3] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, "Simblock: A blockchain network simulator," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 325–329.

[4] M. Basile, G. Nardini, P. Perazzo, and G. Dini, "On improving simblock blockchain simulator," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, 2021, pp. 1–6.

[5] K. Shudo, T. Hasegawa, A. Sakurai, and R. Banno, "Blockchain network studies enabled by simblock," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2023, pp. 1–2.

[6] C. Lee, C. Kang, H. Ko, J. Woo, and J. W.-K. Hong, "A comprehensive and quantitative evaluation method for blockchain protocols," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2023, pp. 1–5.

[7] M. N. Nguyen. (2018) Tanglesimulator. [Online]. Available: https://github.com/minh-nghia/TangleSimulator

[8] M. R. A. Lathif, P. Nasirifard, and H.-A. Jacobsen, "Cidds: A configurable and distributed dag-based distributed ledger simulation framework," *Proceedings of the 19th International Middleware Conference (Posters)*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53721315

[9] M. Zander, T. Waite, and D. Harz, "DAGsim: Simulation of DAG-based distributed ledger protocols," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, pp. 118–121, 01 2019.

[10] B.-Y. Lin, D. Dziubałtowska, P. Macek, A. Penzkofer, and S. Müller, "Tanglesim: An agent-based, modular simulator for dag-based distributed ledger technologies," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2023, pp. 1–5.

[11] S. Y. Popov, "The tangle," 2016.

[12] S. Müller, A. Penzkofer, N. Polyanskii, J. Theis, W. Sanders, and H. Moog, "Tangle 2.0 leaderless nakamoto consensus on the heaviest dag," *IEEE Access*, vol. 10, pp. 105 807–105 842, 2022.

[13] E. Davoodijam, N. Ghadiri, M. Lotfi, and F. Rinaldi, "Multigbs: A multi-layer graph approach to biomedical summarization," *Journal of Biomedical Informatics*, vol. 116, p. 103706, 02 2021.

[14] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2006.

[2]IOTA-core ongoing issues, available at https://github.com/iotaledger/iota-core/issues, 2024.