Performance Analysis of the IOTA DAG-based Distributed Ledger

CAIXIANG FAN, University of Alberta, Canada
SARA GHAEMI, University of Alberta, Canada
HAMZEH KHAZAEI, York University, Canada
YUXIANG CHEN, University of Alberta, Canada
PETR MUSILEK, University of Alberta, Canada

Distributed ledgers provide many advantages over centralized solutions in IoT projects, including but not limited to improved security, transparency, and fault tolerance. To leverage distributed ledgers at scale, their well-known limitation, i.e., performance, should be adequately analyzed and addressed. DAG-based distributed ledgers have been proposed to tackle the performance and scalability issues by design. The first among them, IOTA, has shown promising signs in addressing the above issues. IOTA is an open-source distributed ledger designed for IoT. It uses a directed acyclic graph to store transactions on its ledger, to achieve a potentially higher scalability over blockchain based distributed ledgers. However, due to the uncertainty and centralization of the deployed consensus, the current IOTA implementation exposes some performance issues, making it less performant than the initial design. In this paper, we first extend an existing simulator to support realistic IOTA simulations and investigate the impact of different design parameters on IOTA's performance. Then, we propose a layered model to help the users of IOTA determine the optimal waiting time to resend the previously submitted but not yet confirmed transaction. Our findings reveal the impact of the transaction arrival rate, tip selection algorithms (TSAs), weighted TSA randomness, and network delay on the throughput. Using the proposed layered model, we shed some light on the distribution of the confirmed transactions. The distribution is leveraged to calculate the optimal time for resending an unconfirmed transaction to the distributed ledger. The performance analysis results can be used by both system designers and users to support their decision making.

CCS Concepts: • **Computing methodologies** → **Model development and analysis**; • **Computer systems organization** → **Peer-to-peer architectures**.

Additional Key Words and Phrases: Performance Analysis, Distributed Acyclic Graph, DAG, Distributed Ledger, Internet of Things, IOTA, Quality of Service, Throughput

ACM Reference Format:
Caixiang Fan, Sara Ghaemi, Hamzeh Khazaei, Yuxiang Chen, and Petr Musilek. 2021. Performance Analysis of the IOTA DAG-based Distributed Ledger. *ACM Trans. Model. Perform. Eval. Comput. Syst.* -, -, Article - (September 2021), 21 pages. https://doi.org/10.1145/1122445.1122456

Authors' addresses: Caixiang Fan, caixiang@ualberta.ca, University of Alberta, 116 St & 85 Ave, Edmonton, Alberta, Canada, T6G 2R3; Sara Ghaemi, University of Alberta, 116 St & 85 Ave, Edmonton, Alberta, Canada, T6G 2R3, sghaemi@ualberta.ca; Hamzeh Khazaei, York University, 4700 Keele St, Toronto, Ontario, Canada, M3J 1P3; Yuxiang Chen, University of Alberta, 116 St & 85 Ave, Edmonton, Alberta, Canada, T6G 2R3; Petr Musilek, University of Alberta, 116 St & 85 Ave, Edmonton, Alberta, Canada, T6G 2R3.

2376-3639/2021/9-ART- $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Distributed ledger technologies (DLTs) have obtained a lot of attention from both industry and academia because of some key advantages such as being decentralized, secure and trust-free. In the DLT world, participants are no longer in need of a trusted third party such as banks to complete a transaction, even without trust to each other. The DLTs are essentially distributed databases for storing and sharing data across all nodes in a network. Based on different usage contexts, various types of data, including transaction records (e.g., Bitcoin [31]), contracts [8], and even personal healthcare information [30], can be stored on a distributed ledger (DL) system. According to a recent survey [7], there are about 58 industries (e.g., law enforcement, ride-hailing, and stock trading) that could be transformed by DLTs in the future. Within most DLTs such as blockchain, the data is wrapped into a structure called blocks. Each block includes the cryptographic hash of the prior block, a timestamp and transaction data to create a chain of linked blocks. This way, it is nearly impossible to tamper with data on blockchain because any changes on a single block require to change all the history before it. Clearly, DLT has a great potential to become an effective solution to overcome the data management and security challenges in IT systems. However, various technical details, such as the consensus algorithms and the underlying data structure have a great impact on the performance of DLTs.

Among many types of DLTs, directed acyclic graph (DAG) is considered to be one of the answers to the low latency, high throughput, and scalability challenges in applications such as M2M micro-payment [44]. Within DAG, transactions can directly be attached to a graph without waiting to be wrapped into a block like in a standard blockchain system. Moreover, all newly added transactions can simultaneously run on different chains, making the performance much higher than a single chain. By contrast, traditional blockchain systems, such as Bitcoin [31], Ethereum [6] and Hyperledger [2], first need to put transactions into a block, and then linearly process the block; this will result in low performance. For example, in Bitcoin, it takes on average 10 minutes for a transaction to be confirmed. For the sake of security of large transactions, it is usually recommended that the merchants wait for confirmation of at least six blocks, which implies one hour to complete a transaction. In addition, to avoid forking, the block generation rate is limited in blockchain systems such as Bitcoin and Ethereum, thus dramatically limiting the transaction throughput. To address this issue, researchers have proposed many solutions such as lightning [33], FireLedger [5] and DAG, from different perspectives. Therefore, as one of such solutions, DAG-based DLs under well-designed consensus are theoretically more performant than traditional blockchains.

From the perspective of quality of service (QoS), service-level agreement (SLA), and blockchain as a service (BaaS) [36], the transaction throughput, average waiting time (transaction delay) and system scalability are essential for a DL system and user experience [24]. This is because these metrics reflect the transaction processing capacity, usability and extensibility of a DL system. In particular, transaction throughput, expressed as TPS at the network level, refers to the rate at which valid transactions are stored by the ledger in a defined time period [32]; transaction delay describes a network-wide view of the time taken for a transaction to be valid across the network since it has been issued [32]; and system scalability determines if the ledger can handle a growing amount of TPS as more resources (e.g., CPU, RAM, or nodes) are added to the existing network. As a counterpart to the traditional blockchain, the DAG-based IOTA claims to be scalable and to provide high throughput because of its innovative data structure and efficient consensus design. In our previous work [13], we designed an energy transactive system for smart communities using IOTA and explored the system scalability. Based on our experiments and analyses, we showed that the proposed system was scalable and effective for IoT use cases. In this paper, we focus on the system

performance, including throughput and transaction reattachment waiting time, of DAG-based DL in the same IoT scenario presented in [13].

Here, *confirmation* or *confirmed* means that the transaction gets enough proofs or validations to be trusted and included in the ledger. This is the target state for every normal transaction. *Validation* is an examination process including amount, signature and time check, which is conducted by peers in a peer-to-peer network. We use the terms *"throughput"* and *"CTPS"*; *"validation"*, *"approval"* and *"reference"* interchangeably throughout this paper.

From a system designer's perspective, it is vital to know the transaction processing capacity of the underlying DLT network. In addition, from the users' point of view, it is critical to know about the optimal time that they need to wait before reattaching the transactions, if they have not been confirmed yet. If the waiting time is too short, the premature redundant transactions cause network congestion; if the waiting time is too long, the user experience declines, as does the system throughput. Either way leads to a poor system efficiency. In this paper, we strive to answer two vital research questions about the performance of a private IOTA network:

RQ1. From the system designer's perspective, which factors influence the throughput of an IOTA system? And how, quantitatively, do they impact the throughput?

RQ2. From the user's perspective, what is the optimal waiting time for confirmation before resubmitting the same original transaction?

To address the above questions, we perform the following steps:

(1) Like other empirical analysis approaches [19] [42], we first study the system throughput by leveraging and extending the DAG-based DL simulator for simulating IOTA to identify significant factors such as transaction arrival rate (λ), different TSAs, weighted TSA randomness parameter (α), and network delay reflected by distance (D).
(2) To find a pattern or relationship between the throughput and design parameters, we statistically analyze the performance data obtained from different configurations and parameter settings to identify potential influence factors for both simulations and experiments. Here, experimental data are used to validate the simulation results and then collectively to answer RQ1.
(3) We decompose the transaction confirmations into layers to explore the confirmation process in a fine-granular fashion. This way, we have a better understanding of transaction confirmation time with more details on how confirmations are distributed; provided that, we obtain a reasonable estimate for RQ2.

The key contributions of this paper can be summarized as follows:

- It describes an extension of the DAGsim simulator [45], a simulation tool of DAG-based distributed ledger protocols, to support the currently running consensus on the public IOTA network.
- It provides abundant experimental evidence on how different design parameters influence the IOTA performance and fills a gap in existing research on the optimal reattachment waiting time.
- It proposes a layered model to analyze the confirmed transactions' distribution in IOTA, providing a potential approach to investigate other DAG-based distributed ledgers, such as Byteball [9] and Nano [26].

The remainder of this paper is organized as follows. Section 2 presents the background information on DAG-based distributed ledgers and IOTA. In Section 3, we empirically analyze IOTA's performance through simulations to answer RQ1. In Section 4, we propose an analytical layered model to explore the distribution of confirmed transactions in IOTA and leverage the proposed

model to answer RQ2. The experimental results and main findings are also presented in these two sections. Section 5 gives a brief review of existing work on DL system performance evaluation, including simulation and analytical models. Finally, Section 6 concludes this paper and states some potential future directions of research.

2 BACKGROUND

In this section, we first provide a brief introduction to DAG-based distributed ledgers. Then, we focus on describing the most popular DAG implementation, IOTA.

2.1 DAG Distributed Ledgers

DAG-based distributed ledger, also called DAG distributed ledger or DAG ledger, is a promising DLT that stores transaction data as vertices of a directed acyclic graph. Different new transactions can be appended to different vertices in a DAG at the same time. Compared to blockchain, which bundles transactions in blocks and stores blocks one by one to a chain, DAG ledger naturally obtains better concurrency, as shown in Figure 1. Therefore, it has many advantages over blockchain on transaction throughput, network scalability and resource efficiency.

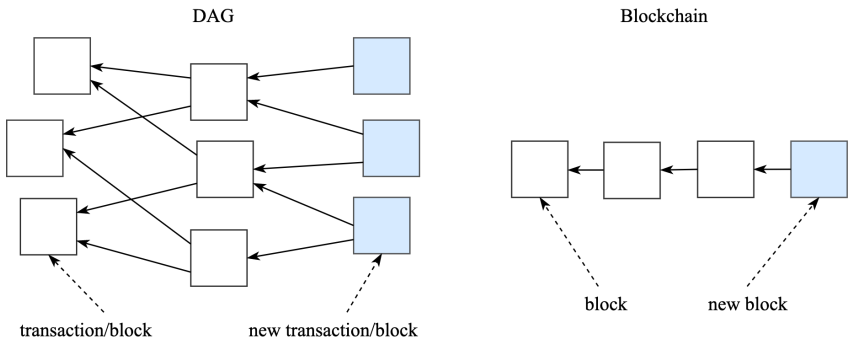


Fig. 1. Structure comparison between DAG and blockchain.

DAG ledger and blockchain are the two most popular DLTs. According to the graph vertex granularity, DAG ledgers can be further divided into two categories: block-based (blockDAG) and transaction-based (TDAG) [44]. The former first wraps transactions into blocks, and then appends the blocks onto a DAG. Some examples are CDAG [16], PHANTOM [39] and SPECTRE [38]. The latter attaches transactions to a DAG immediately and directly, without waiting for block composition. Three notable examples are IOTA [34], Byteball [9] and Nano [26]. Since blockDAG is still in the conceptual stage, we focus our research on TDAG. For simplicity, DAG refers to TDAG throughout this paper without further specification.

In the design of DAG ledgers, each end user issuing a transaction also needs to validate the previous transactions. All participants act as validators and contribute to maintain the network. In other words, no miners are needed in DAG distributed ledger systems. For example, IOTA (see Section 2.2) requires the network participant or node to approve two previous transactions in order to issue a new transaction, while Byteball encourages referencing all unapproved transactions. In addition, multiple transactions can be added to a DAG simultaneously and concurrently to increase system throughput and scalability.

2.2 IOTA

IOTA is a DAG-based open-source value transfer platform designed for the Internet of Things. It is currently the most popular representative of DAG ledgers. This section briefly introduces the basic concepts and consensus mechanism of IOTA.

In IOTA, all transactions are linked together to form a DAG (see Figure 2), called *Tangle*. A transaction is the fundamental operation unit that can stand alone. A *tip* is a newly issued but not approved/validated transaction. Each transaction has its own weight and a cumulative weight. The *weight* reflects the computation resource that a sending node puts into this transaction. In order to issue a new transaction, a node must select two tips to validate. Once the validation is finished, this node will attach the newly issued transaction to the selected tips. This attachment is called a *reference*, through which all transactions are linked together. At the same time, the newly issued transaction adds its own weight to all the predecessors' cumulative weight. For example, all the referenced transactions' (v_1 to v_9) cumulative weights in Figure 2 will add one after v_{10} is attached to the Tangle.

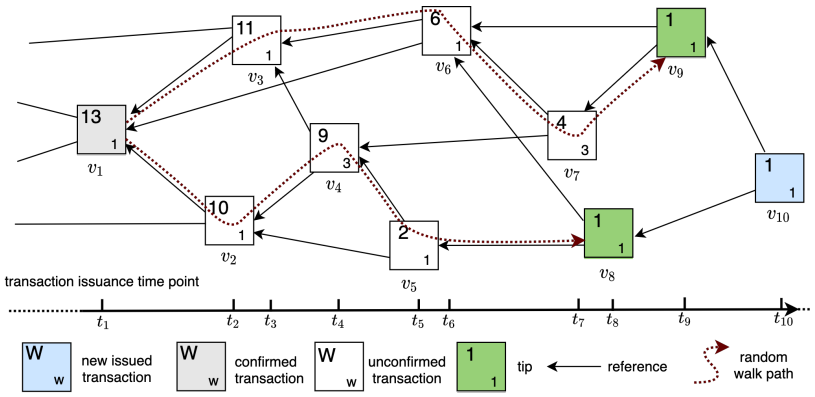


Fig. 2. An example of DAG in IOTA. "w" stands for weight and "W" stands for cumulative weight.

To find a good balance between punishing lazy behavior and not leaving too many tips behind, IOTA recommends a sampling approach named *weighted Markov chain Monte Carlo (MCMC) random walk* or *weighted MCMC* to select two tips. For example, the selected tips in Figure 2 are v_9 and v_8, with the random walk paths $(v_1, v_3, v_6, v_7, v_9)$ and $(v_1, v_2, v_4, v_5, v_8)$, respectively. This approach leverages equation 1 [34] to calculate the probability of choosing the next particular approver for each step in a random walk.

$$P_{xy} = \frac{e^{\alpha H_y}}{\sum_{z:z \rightsquigarrow x} e^{\alpha H_z}} \tag{1}$$

where P_{xy} is the probability to walk from transaction x to y, H_y is the cumulative weight of y, and $z \rightsquigarrow x$ means "z directly approves x". Therefore, in the random walk step, a transaction with higher cumulative weight has a much higher probability to be selected and approved. In other words, the probability of walking from x to y increases exponentially with the cumulative weight of y, multiplied by α. For example, using equation 1, the probability walking from v_1 to v_3 in Figure 2 is calculated as $P_{v_1 v_3} = \frac{e^{11\alpha}}{e^{11\alpha}+e^{10\alpha}+e^{6\alpha}}$. Here, α is a parameter regulating the strength of the bias in the weighted random walk. A higher α-value means that heavier tips are favoured when attaching new transactions to the DAG. If we set α to zero, the random walk is called *unweighted MCMC*. If

we set α to a very high value, we get the superweighted walk. We can also use a uniform random tip selection (URTS) approach to select two from all available tips for comparison.

In IOTA, there are two consensus mechanisms: Coordinator and Coordicide [35]. The former is the currently running consensus, while the latter is still under development. Coordinator is a special participant or node acting as a "finality device" to confirm transactions by periodically generating zero-value transactions, called *milestones*. With Coordinator, IOTA leverages the above discussed weighted (or biased) TSA to deal with conflicts and simply considers a transaction as confirmed if and only if it is referenced by a milestone. In this consensus, an issued transaction will continuously get approved/validated by peer nodes until it eventually gets confirmed. The more references a transaction gets, the more trustable and acceptable it becomes. As such, it has a higher probability to be referenced (indirectly) by the next coming milestone, and thus confirmed. With Coordicide, IOTA proactively resolves conflicts and reaches consensus through a voting mechanism rather than the weighted MCMC TSA [35].

2.3 IOTA Performance Metrics

In this subsection, we introduce two important performance metrics in IOTA, which are our interest of research in this paper.

Throughput. The throughput is defined as the confirmed transactions per second (CTPS), which represents the transaction processing power of the system. As for the definition of confirmation, it depends on the consensus used in the system, see Section 2.2 for details. Particularly, we choose COO as the consensus throughout this study to make it more practical, because this is the currently running consensus in IOTA.

Reattachment Waiting Time. Similar to a blockchain system, it may take seconds or minutes for a transaction to eventually be added to the IOTA ledger. In our system, latency is defined as the time between a transaction's arrival request at a node and its confirmation. Based on this definition, each client wants a lower latency under the same security conditions. Sometimes a transaction may not get confirmed for a long time, causing high latency. It is also possible for a transaction to get abandoned eventually after a long wait. In these cases, the transaction should be *reattached* to a new position in the Tangle. *Reattachment* is the process of issuing the same original transaction to a new position in the Tangle, to increase the confirmation probability and decrease the latency. We define the time between two attachments as the *Reattachment Waiting Time (RWT). Reattachment* requires performing PoW and tip selection again for determining the two new tips to be attached. Therefore, too short RWT will not only waste power, but also cause network congestion due to the number of redundant transactions. On the other hand, a long RWT will dramatically increase confirmation latency and decrease user satisfaction.

3 IOTA PERFORMANCE SIMULATION

The research described in this contribution has been conducted within the context of a private IOTA network designed for smart communities. Two approaches, a simulation and an analytical layered model, are proposed to answer two different performance questions, i.e., system throughput and reattachment waiting time (RWT), respectively. In this section, we focus on IOTA performance simulation, while in the next section (Section 4), we discuss the layered model to explore the confirmation distribution in IOTA.

To conduct the simulation, we leverage and extend the DAGsim simulator [45], which is an asynchronous, continuous time, and multiagent simulation framework for DAG-based distributed ledgers. In addition to the analysis of simulation results, we run experiments in a private IOTA network to validate the results. For better understanding, all symbols and acronyms used for performance analysis are summarized in Table 1.

Table 1. Descriptions of all used symbols and acronyms

Notation	Description
λ	transaction arrival rate
λ_{M}	milestone arrival rate
α	randomness factor for weighted random walk
$agents$	the total number of nodes in a simulated network
d	distance between simulation agents, reflects network delay
D	distances matrix for all agents
$CTPS$	confirmed transactions per second, reflects transaction throughput
RWT	reattachment waiting time
DL	distributed ledger, e.g., blockchain and DAG ledger
DAG	directed acyclic graph, a finite directed graph with no cycles
$Tangle$	DAG structure in IOTA, formed by all connected transactions
COO	coordinator, generates milestones to confirm transactions, also refers to IOTA consensus based on the coordinator
$MCMC$	Markov chain Monte Carlo, a sampling approach for IOTA tip selection
$URTS$	uniform random tip selection
TSA	tip selection algorithm, e.g., weighted/unweighted MCMC and URTS
$IREA$	indirect references extraction algorithm
PoW	proof of work, a protection mechanism asking client to solve a cryptographic puzzle before issuing transactions

To explore the influence of some impact factors on the first IOTA performance metric, transaction throughput, an empirical study is conducted through simulations. These factors include transaction arrival rate λ, TSAs, network latency d and randomness parameter α of the weighted MCMC TSA. For other tested parameters such as network size, PoW difficulty, and the number of Coordinators, refer to our previous work [13] for details. Before presenting the simulations, we first talk about the simulation tool.

3.1 IOTA Simulator Extension

COO consensus, based on the DAGsim simulator [45], has been developed by the authors to perform the simulations. As the original DAGsim only supports consensus without COO, no milestones are generated in between regular transactions. We extended this simulator to support COO consensus by introducing milestones. Technically, we configured the extended DAGsim to generate and broadcast milestones to the network every 60 seconds, just like the currently running IOTA mainnet. The generated milestones are acting exactly like regular transactions, but with the ability to confirm transactions. Our extended version of DAGsim is available publicly[1].

When we want to find out all the confirmed transactions in the simulation data, according to the definition of Coordinator consensus mechanism in Section 2.2, we simply check if a transaction is directly or indirectly referenced by a milestone. In the original DAGsim simulator, for each transaction, we only have access to the transactions that are directly referenced by this transaction in the simulation data. However, we also need indirect references when using COO consensus. To fetch this information, we propose a recursive solution named *Indirect References Extraction Algorithm (IREA)* as shown in Algorithm 1.

[1]https://github.com/pacslab/iota_simulation

Algorithm 1 Indirect References Extraction Algorithm

```
indirect_references = empty
function FIND_REFERENCES(tx, direct_references)
    APVD_1 = The 1st transaction approved by tx
    APVD_2 = The 2nd transaction approved by tx
    if tx is genesis then
        End
    else if tx is in indirect_references then
        Append the new TXs to indirect_references
        End
    else
        if APVD_1 is not in indirect_references then
            Append APVD_1 to indirect_references
        find_references(APVD_1, direct_references)
        if APVD_2 exists then
            if APVD_2 is not in indirect_references then
                Append APVD_2 to indirect_references
            find_references(APVD_2, direct_references)
```

It utilizes a recursive function to find the transactions directly referenced by the input transaction. According to the random walks, we know that there are always two (or at least one if the walks overlap) transactions directly referenced by any issued transaction. These two transactions are added to a list, and the recursive function is run again for each of them. This goes on until the genesis,i.e., the first transaction, is reached or a transaction that is already in the list is encountered. By running this algorithm, all transactions confirmed by a milestone can be found from the simulation data and, subsequently, the throughput (CTPS) can be calculated.

It is worth noting that we extend the DAGsim simulator only from the functional perspective, by simply introducing milestones in simulation and proposing the IREA to identify confirmed transactions. Our goal is to obtain transaction simulation data from this tool. We didn't work on any performance improvement for the simulator itself, which is beyond the scope of this study. Therefore, it is still not efficient in support of large-scale simulations. For example, it takes more than 8 hours to simulate 10,000 transactions on an i7-8700 processor with 16GB RAM.

3.2 Simulation Setup and Results

To collect the simulation data, we run a group of 10 simulations with 6000 transactions, 20 agents, $d = 1$, $\alpha = 0.001$ and λ varying from 1 to 10 with $step = 1$, see Table 2. This is a base-line configuration which we use to explore the influence in comparison with other configurations. Then, we run 5 independent simulations by only changing λ varying from 10 to 30 with $step$ = 5 and transactions from 3,000 to 9,000 with $step = 1,500$ to explore higher rate scenarios. In total, over 90,000 transactions are simulated. In all simulations, λ_{M} is set to be 1/60, i.e. one *Milestone* is issued to the Tangle every minute, because this is the setting in current running IOTA main net; for each simulation, *transactions* is set to 6,000, so that at least 10 *Milestones* (namely 10 replicas) are ensured for each λ configuration. The simulations are conducted on a DELL PC with Windows 10 OS, 8th Generation Intel Core$^{\mathrm{TM}}$ i7-8700 12-Core Processor, and 16GB RAM.

After simulation, the proposed IREA is used to extract the transaction confirmation data and conduct a statistical analysis on the data. The result provides an almost linear relationship between

Table 2. Default parameter configurations

Parameter	Simulation	Configurable	Experiment	Configurable
λ[a]	1~10	✓	1~10	✓
α[b]	0.001	✓	0.001	✓
TSA[c]	MCMC	✓	MCMC	✗
d[d]	1	✓	≈1 ms	✗
agents	20	✓	20	✓
transactions	6000	✓	NA	✗
COOTick	60 s	✓	60 s	✓

[a] 1~10 are explored in both; 15, 20, 25, and 30 are explored in simulation.
[b] 0.001, 0.01 and 0.1 are explored in both simulation and experiment.
[c] weighted MCMC, unweighted MCMC and URTS are explored in simulation.
[d] 1, 5 and 10 are explored in simulation.

CTPS and λ, as shown in Figure 3a, in which all CTPS values are obtained by averaging over all confirmations of 10 replicas.

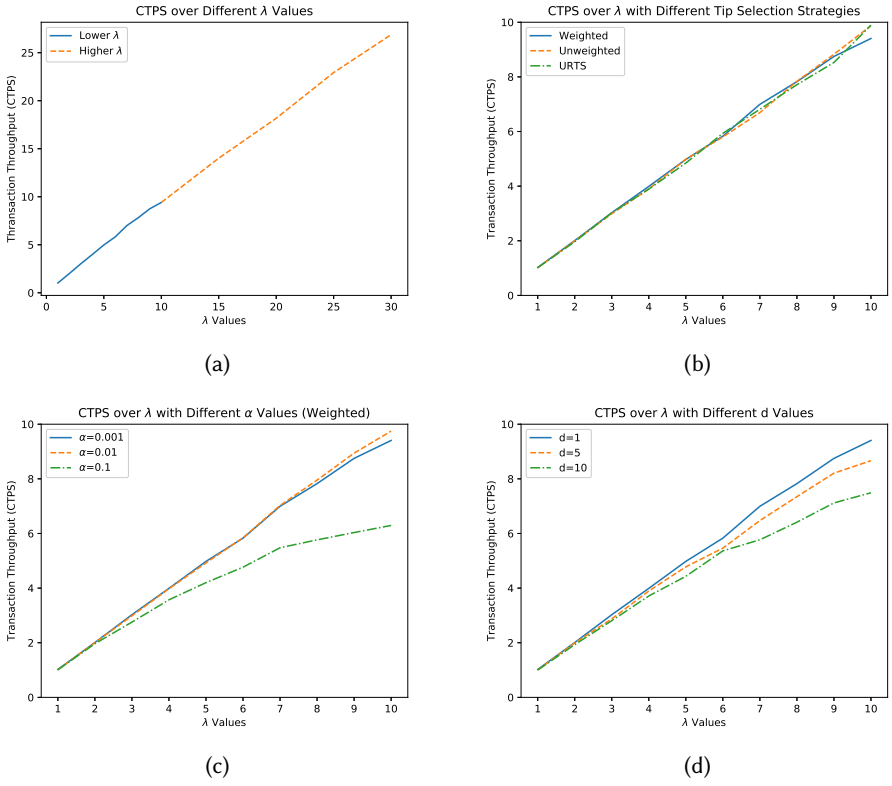


Fig. 3. Simulation results: CTPS over λ under different influence factors (a) λ only (b) TSAs (c) randomness α (d) distance d.

To examine the impact of different TSAs on CTPS, we conduct two more groups of simulations (10 simulations in each group) on the unweighted MCMC and URTS strategies, respectively. In the weighted MCMC random walk, to explore the influence of the randomness factor α on CTPS, we conduct 2 groups of simulations with α=0.01 and 0.1, respectively. In addition, different network delays indicated as distances are examined, i.e., d=1, 5, and 10, respectively. For each group of simulations, the value of λ is varying from 1 to 10 with $step = 1$ (refer to Figure 3b, Figure 3c and Figure 3d).

As can be seen in Figure 3b, there are almost no differences for CTPS under different TSAs, i.e., *weighted*, *unweighted* and *URTS*. This is because when α is set to the default value 0.001, there is sufficient randomness in the tip selection random walk. Nevertheless, the α values have an obvious impact on CTPS in weighted random walk. As can be seen from the Eq. 1, larger α will increase the probability to select heavier tips so that most new coming transactions will always be attached to the heaviest path. It shows that as α increases from 0.001 to 0.01, there is no big difference on CTPS; but when α raises to 0.1, both CTPS and its increase rate decrease dramatically. In a more extreme case, CTPS even keeps flat when α is set to a big value, e.g., 10 [21], which implies a linked chain rather than a DAG in IOTA.

However, the different distances which simulate the network delay of IOTA do not show much difference in general as shown in Figure 3d, which means that network delays have a limited influence on the throughput under the examined situations. This can be explained as follows. On the one hand, compared to the time spent on PoW and transaction validation, network delay only takes a very small portion of the whole transaction life from being sent to getting confirmed. On the other hand, all examined d values remain quite low, which further weakens their influence on the performance.

3.3 Parameter Analysis

To discuss how λ influences IOTA's performance, we model the IOTA system as a single-server network with first-come-first-served (FCFS) servicing policy. The service rate of this network refers to the maximum transaction processing capacity μ, which can be obtained from the load test in practice. In most simulation settings, the actual transaction throughput (denoted as X) keeps a near-linear growth against the transaction arrival rate λ, i.e., $X = k \cdot \lambda$. This trend continues until λ is larger than μ. As transactions are continuously issued and confirmed, the throughput should be the same as the arrival rate when $\lambda < \mu$ in this single-server network model. However, since there is a probability that an issued transaction will be eventually abandoned or orphaned under the random walk TSA, the actual transaction throughput becomes less than λ. In addition, more transactions get abandoned in high load (see Figure 3a). Therefore, λ dominates the growth of the transaction throughput when $\lambda < \mu$, while there is an obvious difference between λ and the actual transaction throughput due to the abandoned transactions. This difference is determined by the transaction attachment mechanism, namely TSA in the Tangle.

Different TSAs affect the growth of Tangle through impacting its properties such as the number of tips, transaction's cumulative weight, and the time until the first approval [22]. The simplest TSA is the URTS, which selects a tip from the set of all available tips in a uniform random manner. Another strategy is MCMC (see 2.2), which is a sampling approach based on a random walk starting from an entry point in the Tangle towards a tip. If the random walk is biased to a more weighted transaction in each step, we call it *weighted MCMC*; if there is no bias in random walks, we call it *unweighted MCMC*. Since the URTS and unweighted MCMC do not encourage to validate more weighted or honest transactions to build a main DAG, they leave vulnerabilities for attackers to build a parasite chain where double spending attacks can be launched (see [22] for details). Therefore, they are not secure against *parasite chain attacks* [22], and should be only treated as tools to study the Tangle

rather than applied in production. To address this problem, IOTA employs the weighted MCMC (see section 2.2) as its TSA to encourage new transactions to verify honest tips and to achieve consensus. The bias level is controlled by a randomness parameter α. Theoretically, the greater the value of α, the greater the probability that the random walk will choose a more weighted tip in each step. Thus, the DAG is more secure by always expanding on the main chain. However, there are more abandoned honest transactions in this case, which will impact the transaction throughput.

We can observe that the Tangle performs almost the same on throughput under URTS and MCMC with $\alpha = 0, 0.01$ in Figure 3b, as well as the weighted MCMC with $\alpha = 0.001$ in Figure 3c. They all have sufficient randomness to ensure that every tip has a chance to be selected, leaving a smaller number of abandoned transactions in the Tangle. However, the throughput drops significantly when $\alpha = 0.1$ in Figure 3c, where more honest transactions are abandoned because of the biased tip selection strategy. Therefore, it is an interesting research topic to find the optimal α, or propose more TSAs (e.g., hybrid TSA [14] and first order biased random walk [10]) to tackle the trade-off problem between security and efficiency of the Tangle. In addition, the time complexity of URTS is $O(n)$, while both weighted and unweighted random walk TSAs have $O(n^2)$ time complexity.

Even though the network delay has a very limited influence on throughput compared to on transaction latency, we can observe a drop on the throughput as the delay increases in Figure 3d. In the IOTA network, a node relies on TCP/IP to broadcast transactions and synchronize the local ledger with its neighbors. High network delay will cause an asynchronization problem among different local ledgers on the nodes. This asynchronization will further decrease the number of new tips from a delayed node's view. Since the weighted MCMC prefers new tips, more delayed transactions will get older and eventually abandoned as the network delay increases. Thus, the transaction throughput decreases.

3.4 Experimental Validation for Simulation

To validate the simulation , we conduct three groups of experiments on a medium-sized network, with the α configurations of 0.001, 0.01, and 0.1. For each α, the total λ values vary from 1 to 10 with $step = 1$, with other parameters remaining default as shown in Table 2. We employ IOTA Implementation Reference[2] (IRI 1.6.1) with Docker to deploy a private network of 20 nodes on the SAVI OpenStack cloud platform[3]. Each node is a virtual machine with the flavor of medium size, see Table 3 for configuration details. The open-source Compass[4] provided by the IOTA Foundation is used as the COO to generate milestones and confirm transactions in the Tangle. The COO is set to generate a milestone every 60 seconds, just as the simulations.

Table 3. Experimental environment

Network Nodes	Number	CPU	RAM	Disk
IRI Node	20	2 VCPU	4GB	40GB HD
Client Node	1	4-Core i5	8GB	256GB SSD

To better control the transaction arrival rate λ and avoid the impact of PoW to λ, we bring all PoW to a PC client with the configurations shown in Table 3. We run all experiments with the transaction requests in a Poisson process, i.e., the transaction interarrival time follows an

[2]https://hub.docker.com/r/iotaledger/iri
[3]https://www.savinetwork.ca
[4]https://github.com/iotaledger/compass

exponential distribution. The socket ZeroMQ[5] is employed to listen and receive transaction data. To simplify the problem, we only send zero-value transactions.

In total, 151,104 confirmed out of around 169,701 generated transactions are collected from the three groups of experiments. More details on the experimental and simulation results can be found in [11]. Then, we compare the experimental data with the previous simulation data on λ and α to examine the validity and accuracy of the simulation.

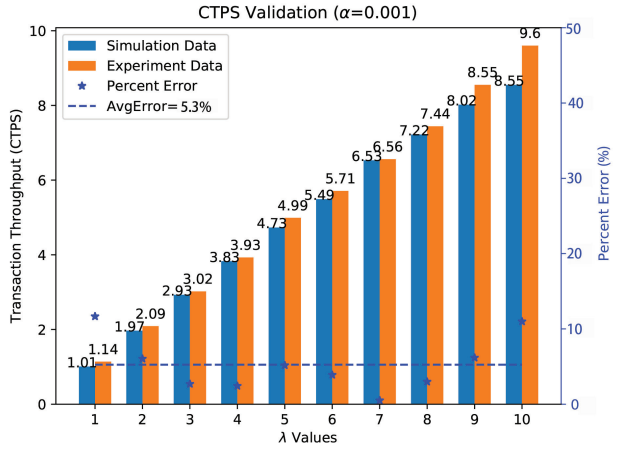


Fig. 4. Experimental and simulation results comparison: λ only.

First, we compare the CTPS values of the simulation and experiment under varying λ 1~10 by keeping other parameters as default (see Table 2) to calculate the percent error. As we can see in Figure 4, (1) both the simulation and experimental CTPS results increase almost linearly as λ values increase, implying a good scalability; (2) they match well to each other with an average percent error of 5.3%. Here, the percent error is calculated from the following equation,

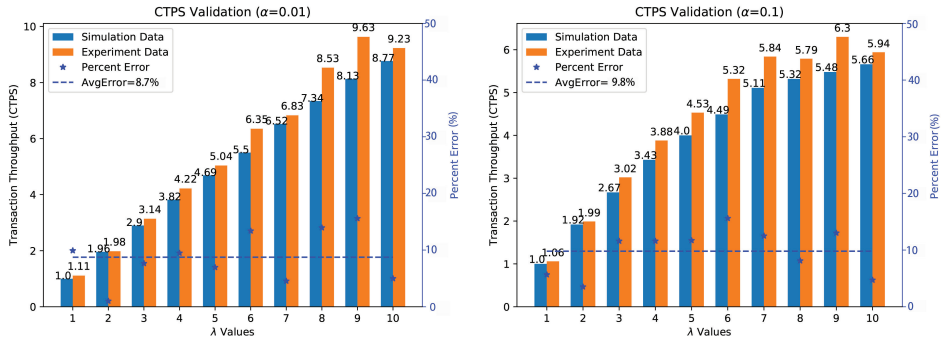
$$Percent\ Error = \frac{|CTPS_{\text{exp}} - CTPS_{\text{simu}}|}{CTPS_{\text{exp}}} * 100\% \tag{2}$$

Second, we compare the simulation and experimental results in terms of different α values. As can be seen in Figure 5, simulation and experiment are well in tune for the two values of α. The average errors are 8.7% and 9.8%, respectively. Having the average errors below 10%, we assume that the simulation data can effectively and accurately capture the Tangle behaviour in the real world.

4 ANALYTICAL LAYERED MODEL

The previous simulation explored some CTPS influential factors, identified the most important one, and provided a quantitative relationship between CTPS and λ (i.e., linear relationship). However, it is difficult to describe more details such as how these confirmations are distributed in the Tangle, which keeps the second question about reattachment waiting time still unanswered. Therefore, we propose a layered model to explore the confirmation distribution in each single graph layer.

[5]https://docs.iota.org/docs/client-libraries/0.1/how-to-guides/python/listen-for-transactions



(a) (b)

Fig. 5. Experimental and simulation results comparison under different α values: (a) α=0.01 (b) α=0.1.

4.1 Model Description and Solution

In the IOTA Tangle, we define a layer as all confirmed transactions with the same depth from a Milestone in a hierarchical architecture, as shown in Figure 6. In the case of two different transactions referencing the same transaction with different depths, we take the minimum layer index as the layer depth for this transaction. For example, in Figure 6, transaction 5 holds references from both 1 and 4, which are from different layers, $Layer_1$ and $Layer_2$. In this case, we assume that 5 is located in $Layer_2$ rather than $Layer_3$. With respect to this layering decomposition and using the previously mentioned simulation data set with over 90,000 transactions, we extract the transaction confirmation number in each layer of the DAG, as shown in Figure 6.

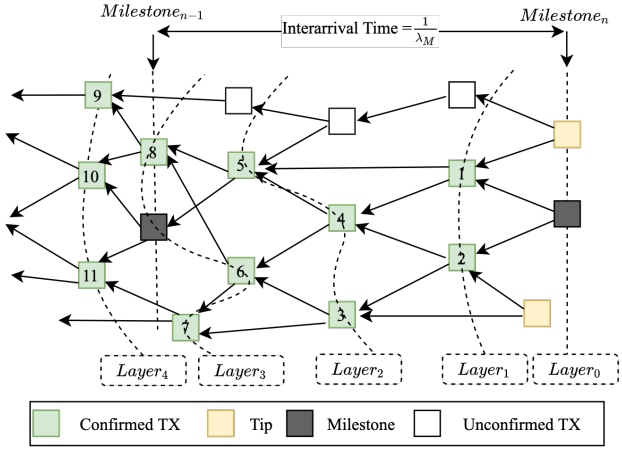


Fig. 6. Layered model for transaction confirmations.

After plotting the confirmed transactions over the layer number for each λ, we observe a lot of bell-shaped curves, such as the blue dot "Data" curve shown in Figure 7, which points us to the nonlinear model fitting, e.g., Gaussian model. Therefore, after taking the average confirmations of all milestones for each λ, we strive to fit our simulation data as a nonlinear model to characterize the relationship.

In total, for each λ we use 45 nonlinear models to fit our data in CurveExpert[6]. The results show that under all λ values except for λ=1, the Gaussian Model outperforms others and is always listed in top 3 models, as we can see from the example of λ=10 in Figure 7. In our fitting, we use the target data set under various λ values by taking the mean layered transactions of milestone 5, 6, 7, 8, and 9, by getting rid of the potential warming-up and cooling-down phases. The fitting results of Gaussian Model are listed in Table 4.

Table 4. Gaussian model fitting results for different λ values

λ	1	2	3	4	5	6	7	8	9	10
Correlation Coefficient	0.934	0.988	0.987	0.982	0.992	0.997	0.984	0.996	0.990	0.991
Standard Error	0.864	0.909	1.274	2.271	1.911	1.476	3.469	2.242	3.733	4.146
a	7.598	16.561	22.439	32.087	40.232	49.378	51.804	66.691	71.021	81.009
b	7.944	8.360	8.976	9.172	9.726	9.390	10.169	9.745	10.298	9.722
c	4.246	3.517	3.742	3.656	3.296	3.298	3.859	3.296	3.536	3.283
AMUB*	13.600	15.000	15.600	16.800	16.400	16.200	17.600	17.000	17.400	16.600
CIUB+	16.265	15.252	16.310	16.337	16.187	15.854	17.733	16.205	17.229	16.156

*Actual Mean Upper Bound, +Confidence Interval Upper Bound

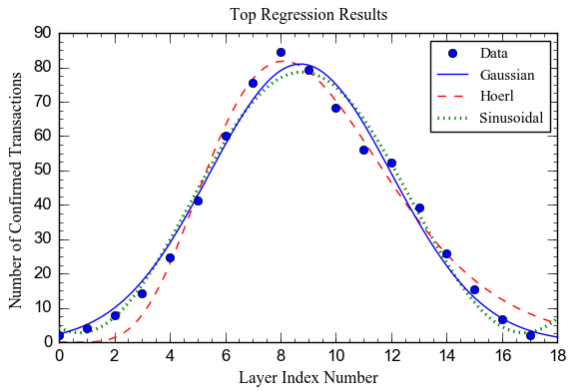


Fig. 7. Simulation data and fitted models for λ=10.

By checking the values of *Correlation Coefficient*, we carefully claim that the mean confirmed transactions located at different layers can be fitted as a Gaussian Model. Therefore, we have the number of confirmed transactions to be a Gaussian function of layer x,

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \tag{3}$$

Here, b has an almost linear increase trend as λ increases, while c almost remains the same from our simulation data in Table 4. This indicates that all examined Gaussian models have a very similar shape, and the next randomly confirmed transaction is expected to be located at a deeper layer in the Tangle with higher λ values. Theorem 1 summarizes our findings.

[6]https://www.curveexpert.net

Theorem 1. *If a transaction remains unconfirmed more than $2/\lambda_M$ seconds after being submitted, the probability that it gets confirmed afterward is sufficiently low.*

Proof. Let layer index be the discrete time dimension, then the *Confidence Interval (CI)* of Gaussian models can be used to estimate the length of time to wait before reattaching transactions. First, let us look at the *Upper Bound* layers in our model. If we take a *CI* of 95%, the *critical value (Z-value)* for this *CI* is 1.96, where $(1 - 0.95)/2 = 0.025$. In our case, this means that there is a very small probability (2.5%) for a confirmation to happen after a specific *Upper Bound* layer. The layer of *CIUB* shown in Table 4 is calculated by the following estimation formula.

Given that

$$Z = \frac{X - b}{c} \tag{4}$$

we have

$$X_{\text{CIUB}} = Zc + b \tag{5}$$

Then, we translate the *Upper Bound* layer to the time dimension by analyzing the layered model. As we notice from Figure 7, the decreasing happens just after a specific layer. From Figure 6, we know that when the confirmation layers of $Milestone_{\text{n}}$ crosses the arrival time of $Milestone_{\text{n-1}}$, there will be a decrease in the number of transactions confirmed by $Milestone_{\text{n}}$ because of the overlap. For example, as shown in Figure 6, transaction 10 and 11 will not be counted as confirmations by $Milestone_{\text{n}}$ since they had already been confirmed by $Milestone_{\text{n-1}}$. Therefore, we empirically notice that the peak CTPS layer in confirmation Gaussian Model refers to the previous *Milestone* arrival time, which is $1/\lambda_M$ seconds ago from the latest one. Thus, the optimal waiting time before reattaching for users is estimated to be around $\mathbf{2}/\lambda_M$ seconds.

4.2 Model Validation

To validate the deductive results of our layered model, we use three groups of experimental data with different α values to conduct a statistical analysis. First, we determine all identical transactions by matching their hash values with the sent and confirmed records, respectively. Then, we leverage the $TimeStamp$ property to calculate the time difference from sending to getting confirmed for each transaction. Since some confirmed transactions recorded in a time period are generated from the previous time segment and cannot be matched within the corresponding sent records, the number of matched transactions is usually less than all actually confirmed transactions. For example, there are 40,023 sent, 39856 confirmed, and 39462 matched transactions when $\alpha = 0.001$.

In total, there are 110,726 confirmed transactions with confirmation waiting time (i.e., they are matched and have the sent-confirmed time difference). Within these transactions, we find that only 4,948 are confirmed after 2 minutes, see Table 5.

Table 5. Statistics of confirmations over 2 minutes ($\mathbf{2}/\lambda_M$ seconds)

α	Conf txs	Conf txs(>2 mins)	Percentage
0.001	39,462	2,235	5.7%
0.01	39,355	2,523	6.4%
0.1	31,909	190	0.6%

The detailed transaction confirmation distribution over time under different α values can be observed from Figure 8. As we know that λ_M is set to be 1/60, i.e., a milestone is issued every minute, so $\mathbf{2}/\lambda_M$ seconds are exactly 2 minutes in our experiments. Therefore, we have only 5.7%, 6.4% and

0.6% of the transactions confirmed after the time of $\mathbf{2}/\lambda_M$ for α=0.001, 0.01 and 0.1, respectively, which is relatively low and well matched with the prediction of our proposed layered model.

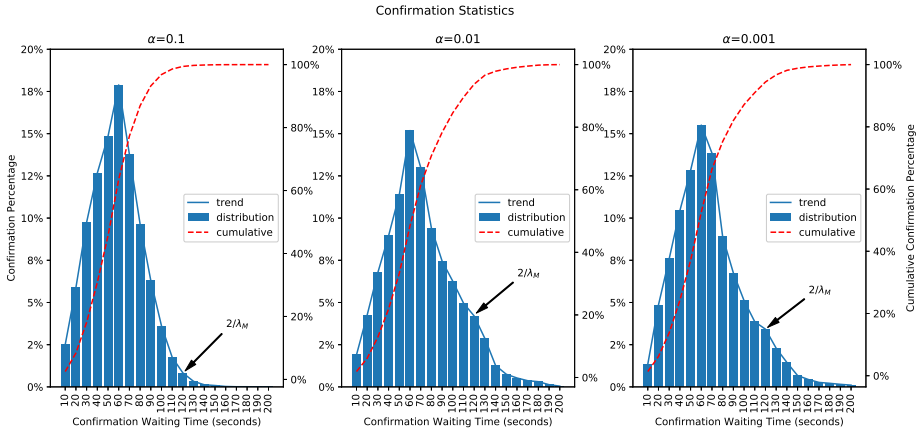


Fig. 8. Experimental confirmation statistics in different time segments; the confirmation ratios are sufficiently low after the times of $2/\lambda_M$.

5 RELATED WORK

This section reviews the related work on the performance evaluation and analysis of both blockchain and DAG-based DL systems, which can be either public or private. In particular, we conduct a brief literature review on leveraging simulations and analytical models (e.g., Markov chains [17]) to analyze DLT's performance.

5.1 DLT Simulations

To explore the block creation performance under PoW consensus algorithm of blockchain, Alharby et al. [1] proposed BlockSim as a framework to simulate discrete events in blockchain systems. This simulator was helpful to understand the details in the block generation process and PoW. However, the authors left the defined test cases validation and verification for future work. Therefore, it is hard to tell if the model is error-free and the simulator semantically works as intended. Similarly, in blockchain-based systems, Yasaweerasinghelage et al. [43] showed the feasibility of using architectural performance modelling and simulation tools to predict the system latency. With a relatively high prediction accuracy of over 90%, the authors discussed how to leverage this simulation to support architectural decision-making, especially on performance in blockchain-based system design.

As for DAG simulations, two IOTA Foundation white papers [20] and [23] built a discrete model and a continuous-time model for IOTA, respectively. The former gave a first glance of IOTA by introducing a discrete model and discussing the relationship between the cumulative weights of transactions and tip numbers over discrete time steps. The simulation results revealed that the number of tips $L(t)$ as a function of time remained stable under the random tip selection strategy. By contrast, MCMC guided tip selection strategy was stable only for small α (0.001) in the examined time intervals. The later paper [23] provided a continuous-time simulation model to validate the analytical prediction about the number of tips $L(t) = \frac{k}{k-1}\lambda h$, which was initially proposed by Popov [34]. The authors also explored the cumulative transaction weights and found

that there was a non-negligible probability of transactions being left behind for larger values of α. For the simulator design, they generalized the tip selection number to be k rather than 2 and chose $3k + 4$ particles to ensure k distinct tips selected from random walks. Each walk starting position was chosen randomly (with uniform distribution) from transactions issued between 100λ and 200λ transactions before. According to the simulation results, they empirically concluded that any starting position placed further than 10λ to 20λ provided the same growth of the number of tips, and it would not influence the tips number. These empirical results could provide directions for further study to improve the simulation efficiency. However, this work did not directly examine or analyze any specific system level performance metrics. In [21], the authors formalized and numerically estimated the probability that a given transaction will be left behind, which means that it never gets confirmed eventually. Through a lot of simulations and data fitting, the authors estimated the probability of being left behind as a function of λ and α, $POBLB(\alpha, \lambda) = a \cdot exp(\frac{b}{\alpha} + \frac{c}{\lambda})$ [21]. However, this function does not take the time factor into consideration, which is very significant to any computer system end users in practice.

To illustrate and visualize the Tangle, Gal [15] developed an IOTA visualization simulator. Through this simulator, one could intuitively observe different tangles generated under various TSAs such as uniform random, weighted and unweighted random walk. In the weighted random walk, the simulator could show how the model parameters such as arrival rate λ and the built-in randomness factor α change the Tangle's shape and the transaction confirmation ratio. However, the total transaction number in this simulator was limited to 500 and the performance metrics were not quantified.

To break out this limitation, Lathif et al. [25] proposed a configurable and interactive DAG-based DLT simulation framework named CIDDS [25] to enable large-scale simulations at the thousand nodes level. Moreover, Bottone et al. [4] presented and developed an extendable multiagent simulator, in which the authors employed NetLogo to provide a 3D visualization of the Tangle.

Similar to the previously mentioned BlockSim [1], Zander et al. [45] proposed and developed a simulator named *DAGsim*, aiming to simulate the DAG-based DL systems. Different from the BlockSim, DAGsim turned to a new type of data structure and focused on the simulation of the IOTA [34] protocol. In their work, the authors presented and implemented an asynchronous, continuous-time, and multi-agent simulator for DAG-based cryptocurrencies. By modelling both honest and semi-honest actors [45], the simulations showed that the agents with lower latency and a higher connection degree had a higher probability of having their transactions accepted in the network. This open-sourced simulator was an efficient tool for simulating different parameters and for understanding the IOTA consensus. However, it faced the same difficulties as other DAG-based DL systems: 1) the starting point for each tip selection random walk, and 2) the cumulative weights update for each transaction. It was very time-consuming when walking from genesis and updating weights transaction by transaction, with a run-time complexity of $O(n^3)$ in the implementation. Therefore, it was very inefficient and not suitable for large-scale simulations, for instance, more than 10,000 transactions. In addition, this research work did not go further into directly evaluating performance metrics rather than exploring the transaction attachment probabilities of each node. Our work borrows this simulator and builds into it the Coordinator confirmation consensus to conduct a performance simulation. Furthermore, we leverage the extended simulator to simulate IOTA for conducting the performance study by setting different configurations.

5.2 Analytical Models for DLT

Sukhwani et al. [41] modeled the Practical Byzantine Fault Tolerance (PBFT) consensus process using Stochastic Reward Nets (SRN). Through this model, they analytically calculated the mean

consensus achieving time for a network of 100 peers. A blockchain network using IBM Bluemix service with production-grade IoT application was created, from which the collected data was used to parameterize and validate the models. Moreover, a sensitivity analysis over a variety of system parameters was conducted to examine the performance of larger networks. With extensive empirical analyses on two releases of Hyperledger Fabric, H. Sukhwani concluded in [40], that the presented Stochastic models could be used to develop Fabric Network management infrastructure.

Li et al. [27] proposed a queuing theory of blockchain systems and provided system performance evaluation. Specifically, a Markovian batch-service queuing system with two different service stages, mining process and the building of a new blockchain, were designed. By using the matrix-geometric solution, they got a stable system condition and analyzed three key performance metrics including mean transaction number in the queue, mean transaction number in a block, and mean transaction confirmation time. Finally, the authors used numerical examples to validate the proposed analytical model. In short, this paper described a clear queue problem and offered a solution based on the commonly used Markovian chain approach. As mentioned in their conclusion, this analytical model had the potential to open a series of promising research in the queuing theory of blockchain systems, even though the proposed model was simple [27].

Motivated by this work, Saulo [37] built a simple M/G/1 queuing model to study the blockchain delays in the Bitcoin network. Transaction delay was one of the most important performance metrics, especially for user experience. The proposed model related the delay of transactions with the time between block confirmations and could be easily parameterized using real measurements. By analyzing the delay from the Little's law and the standard M/G/1 queuing model [18], they found a simple relationship between E(B)-mean time between block confirmation, E(M)-mean number of active blocks, and E(S)-active time of a block; and a relationship between E(D)-mean delay incurred by a typical user transaction and E(B), respectively. This way, they could answer the following two questions: 1) whether a transaction will be confirmed after being seen; and 2) what are the important factors that contribute to the delay of transaction confirmation.

Recently, Memon et al. [29] proposed a queuing theory-based model by segregating the blockchain network into two types of pools, named Memory-Pool and Mining-Pool, which were in charge of handling unconfirmed transactions and mining, respectively. Based on this segregation, M/M/1 and M/M/c queues were used to model the system. In addition, a simulation model developed in Java Modelling Tools was employed to study the blockchain system behavior and its performance metrics such as *Number of Transactions per block*, *Mining Time of Each Block*, *Number of Transactions per Second* and *Waiting Time in Memory-pool*. Similarly, [28] introduced Markov chain model to investigate the impact of different network loads on the performance of DAG consensus process (e.g., Tangle) in terms of cumulative weight growth rate and confirmation time.

More performance modelling approaches for DLTs have been discussed in detail in our recently published survey [12]. The related publications reviewed above provide different perspectives to explore the system performance of DAG-based IOTA. However, neither approach is perfect as analytical models sometimes give high prediction errors if tested with improper parameters, while simulations may encounter resource limitation problems. A good idea is to combine the above models to provide a hybrid solution [3]. In addition, none of the reviewed articles focused on the user waiting time in a DAG-based distributed ledger system. It is another contribution of this paper to fill this gap.

6 CONCLUSION AND FUTURE WORK

In this paper, we have studied the performance of a private IOTA network by experiments, simulations, and a layered model. We leverage these approaches to answer two research questions on

throughput and RWT, with a high level of confidence. In particular, we extended the DAGsim simulator and used it to empirically analyze the influence of transaction arrival rate λ, *TSAs*, randomness α in weighted random walk algorithm and network delay d on CTPS. Among all these impact factors, λ is the most important one, which has a near-linear relationship with the CTPS. Moreover, we leveraged the proposed analytical layered model to explore the confirmation distribution and found that the confirmations are normally distributed in DAG layers, which led to characterizing the Gaussian Model. Using this model, we estimated the RWT for a private IOTA network, which was validated by our experimental results. In conclusion, our extended DAG ledger simulator and the proposed layered model provide valuable insights into the performance of IOTA distributed ledger in private network scenarios.

As for the future work, on one hand, we would like to study how other factors such as encryption algorithms and ledger databases influence the throughput. It would be also interesting to evaluate IOTA's performance under consensus without COO in further studies. On the other hand, our extended simulator did not resolve the efficiency problem in large-scale simulations, which would be another direction of our future research.

ACKNOWLEDGMENTS

The financial support provided by the Future Energy Systems under the Canada First Research Excellence Fund (CFREF) at the University of Alberta is gratefully acknowledged. The authors would also like to thank SAVI cloud and Cybera, Alberta's not-for-profit technology accelerators that support this research through their cloud services.

REFERENCES

[1] Maher Alharby and Aad Van Moorsel. 2018. BlockSim: A Simulation Framework for Blockchain Systems. *ACM SIGMETRICS Perform. Eval. Rev.* 46, 3 (2018), 135–138. https://doi.org/10.1145/3308897.3308956

[2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference (EuroSys '18)*. Association for Computing Machinery, New York, NY, USA, Article 30, 15 pages. https://doi.org/10.1145/3190508.3190538

[3] Janki Bhimani, Ningfang Mi, Miriam Leeser, and Zhengyu Yang. 2019. New Performance Modeling Methods for Parallel Data Processing Applications. *ACM Trans. Model. Comput. Simul.* 29, 3, Article 15 (June 2019), 24 pages. https://doi.org/10.1145/3309684

[4] Michele Bottone, Franco Raimondi, and Giuseppe Primiero. 2018. Multi-agent based simulations of block-free distributed ledgers. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, New York, NY, USA, 585–590.

[5] Yehonatan Buchnik and Roy Friedman. 2019. FireLedger: a high throughput blockchain consensus protocol. *arXiv preprint arXiv:1901.03279* (2019).

[6] Vitalik Buterin. 2014. A next-generation smart contract and decentralized application platform. *white paper* 3 (2014), 37. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf

[7] CBinsights. 2021. *Banking Is Only The Beginning: 58 Big Industries Blockchain Could Transform.* Technical Report. CB Information Services, Inc. https://www.cbinsights.com/research/industries-disrupted-blockchain/

[8] Konstantinos Christidis and Michael Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things. *Ieee Access* 4 (2016), 2292–2303.

[9] Anton Churyumov. 2016. Byteball: A decentralized system for storage and transfer of value. *White Paper* (2016), 1–49. https://obyte.org/Byteball.pdf

[10] A Cullen, P Ferraro, C King, and R Shorten. 2020. On the Resilience of DAG-based Distributed Ledgers in IoT Applications. *IEEE Internet of Things Journal* 7, 8 (2020), 7112–7122.

[11] Caixiang Fan. 2019. *Performance Analysis and Design of an IoT-Friendly DAG-based Distributed Ledger System.* Master of Science Thesis. University of Alberta.

[12] Caixiang Fan, Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. 2020. Performance Evaluation of Blockchain Systems: A Systematic Survey. *IEEE Access* 8 (2020), 126927–126950.

[13] Caixiang Fan, Hamzeh Khazaei, Yuxiang Chen, and Petr Musilek. 2019. Towards A Scalable DAG-based Distributed Ledger for Smart Communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, New York, NY, USA, 177–182. https://doi.org/10.1109/WF-IoT.2019.8767342

[14] Pietro Ferraro, Christopher King, and Robert Shorten. 2018. IOTA-based directed acyclic graphs without orphans. *arXiv preprint arXiv:1901.07302* (2018), 1–11.

[15] Alon Gal. 2018. The Tangle: an Illustrated Introduction. https://blog.iota.org/the-tangle-an-illustrated-introduction-4d5eae6fe8d4

[16] Himanshu Gupta and Dharanipragada Janakiram. 2019. CDAG: A Serialized blockDAG for Permissioned Blockchain. *arXiv preprint arXiv:1910.08547* (2019), 1–13.

[17] Khazaei Hamzeh, Jelena Misic, and Vojislav B. Misic. 2012. Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems. *IEEE Trans. PARALLEL Distrib. Syst.* 23, 5 (2012), 936–943. https://doi.org/10.1109/TPDS.2011.199

[18] Mor Harchol-Balter. 2013. *Performance modeling and design of computer systems: queueing theory in action.* Cambridge University Press, Cambridge, UK.

[19] Alexey Ilyushkin, Ahmed Ali-Eldin, Nikolas Herbst, André Bauer, Alessandro V. Papadopoulos, Dick Epema, and Alexandru Iosup. 2018. An Experimental Performance Evaluation of Autoscalers for Complex Workflows. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3, 2, Article 8 (April 2018), 32 pages. https://doi.org/10.1145/3164537

[20] B Kusmierz. 2017. The first glance at the simulation of the Tangle: discrete model. *IOTA Found. White Paper* (2017), 1–10.

[21] Bartosz Kusmierz and Alon Gal. 2018. Probability of being left behind and probability of becoming permanent tip in the tangle v0.2. *White Paper* (2018), 1–15.

[22] Bartosz Kusmierz, William Sanders, Andreas Penzkofer, Angelo Capossele, and Alon Gal. 2019. Properties of the Tangle for uniform random and random walk tip selection. In *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, New York, NY, USA, 228–236.

[23] Bartosz Kusmierz, Philip Staupe, and Alon Gal. 2018. *Extracting tangle properties in continuous time via large-scale simulations.* Technical Report. working paper.

[24] Murat Kuzlu, Manisa Pipattanasomporn, Levent Gurses, and Saifur Rahman. 2019. Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In *2019 IEEE international conference on blockchain (Blockchain)*. IEEE, New York, NY, USA, 536–540.

[25] Mohamed Riswan Abdul Lathif, Pezhman Nasirifard, and Hans-Arno Jacobsen. 2018. CIDDS: A Configurable and Distributed DAG-based Distributed Ledger Simulation Framework. In *Proceedings of the 19th International Middleware Conference (Posters)*. ACM, New York, NY, USA, 7–8.

[26] Colin LeMahieu. 2018. Nano: A feeless distributed cryptocurrency network. *White Paper* (2018), 1–8. https://content.nano.org/whitepaper/Nano_Whitepaper_en.pdf

[27] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chang. 2018. Blockchain queue theory. In *International Conference on Computational Social Networks*. Springer, New York, NY, USA, 25–40.

[28] Yixin Li, Bin Cao, Mugen Peng, Long Zhang, Lei Zhang, Daquan Feng, and Jihong Yu. 2019. Direct Acyclic Graph based Blockchain for Internet of Things: Performance and Security Analysis. *arXiv preprint arXiv:1905.10925* (2019).

[29] Raheel Memon, Jian Li, and Junaid Ahmed. 2019. Simulation Model for Blockchain Systems Using Queuing Theory. *Electronics* 8, 2 (2019), 234. https://doi.org/10.3390/electronics8020234

[30] Matthias Mettler. 2016. Blockchain technology in healthcare: The revolution starts here. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, New York, NY, USA, 1–3.

[31] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. *White Paper* (2008), 1–9. https://bitcoin.org/bitcoin.pdf

[32] Hyperledger Performance and Scale Working Group. 2018. Hyperledger Blockchain Performance Metrics. *White Paper* (2018), 1–17.

[33] Joseph Poon and Thaddeus Dryja. 2016. The bitcoin lightning network: Scalable off-chain instant payments.

[34] Serguei Popov. 2016. The tangle. *White Paper* (2016), 1–28. http://www.descryptions.com/Iota.pdf

[35] Serguei Popov, Hans Moog, Darcy Camargo, Angelo Capossele, Vassil Dimitrov, Alon Gal, Andrew Greve, Bartosz Kusmierz, Sebastian Mueller, Andreas Penzkofer, et al. 2020. The Coordicide. *White Paper* (2020), 1–55. https://files.iota.org/papers/20200120_Coordicide_WP.pdf

[36] Mayra Samaniego and Ralph Deters. 2016. Blockchain as a Service for IoT. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, New York, NY, USA, 433–436.

[37] Ricci Saulo, Ziviani Artur, Ferreira Eduardo, Souza Jose Eduardo, Menasche Daniel Sadoc, and Alex Borges Vieira. 2018. Learning Blockchain Delays : A Queueing Theory Approach. *ACM SIGMETRICS Perform. Eval. Rev.* 46, 3 (2018), 122–125.

[38] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. 2016. Spectre: Serialization of proof-of-work events: Confirming transactions via recursive elections. *White Paper* (2016), 1–66. https://eprint.iacr.org/2016/1159.pdf

[39] Yonatan Sompolinsky and Aviv Zohar. 2018. PHANTOM and GHOSTDAG: A Scalable Generalization of Nakamoto Consensus. *White Paper* (2018), 1–14. https://eprint.iacr.org/2018/104.pdf

[40] Harish Sukhwani. 2018. *Performance Modeling & Analysis of Hyperledger Fabric (Permissioned Blockchain Network).* Doctor of Philosophy Thesis. Duke University.

[41] Harish Sukhwani, José M Martínez, Xiaolin Chang, Kishor S Trivedi, and Andy Rindos. 2017. Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS).* IEEE, New York, NY, USA, 253–255.

[42] Cheng Wang, Qianlin Liang, and Bhuvan Urgaonkar. 2018. An Empirical Analysis of Amazon EC2 Spot Instance Features Affecting Cost-Effective Resource Procurement. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3, 2, Article 6 (March 2018), 24 pages. https://doi.org/10.1145/3164538

[43] Rajitha Yasaweerasinghelage, Mark Staples, and Ingo Weber. 2017. Predicting latency of blockchain-based systems using architectural modelling and simulation. In *2017 IEEE International Conference on Software Architecture (ICSA).* IEEE, New York, NY, USA, 253–256.

[44] Kimchai Yeow, Abdullah Gani, Raja Wasim Ahmad, Joel JPC Rodrigues, and Kwangman Ko. 2017. Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues. *IEEE Access* 6 (2017), 1513–1524.

[45] Manuel Zander, Tom Waite, and Dominik Harz. 2018. DAGsim : Simulation of DAG-based distributed ledger protocols. *ACM SIGMETRICS Perform. Eval. Rev.* 46, 3 (2018), 118–121. https://doi.org/10.1145/3308897.3308951