SCALABILITY ANALYSIS OF IOT-DAG DISTRIBUTED LEDGERS USING PREFERENTIAL ATTACHMENT TOPOLOGY: A SIMULATION APPROACH

Peter Kimemiah Mwangi [1], Stephen T. Njenga [2] and Gabriel Ndung'u Kamau [2]

[1] Department of Information Technology, School of Computing Information Technology, Murang'a University of Technology, Murang'a, Kenya
[2] Department of Computer Science, School of Computing Information Technology, Murang'a University of Technology, Murang'a, Kenya

ABSTRACT

Directed Acyclic Graph (DAG) based Distributed Ledger Technologies (DLTs) are being explored to address the scalability and energy efficiency challenges of traditional blockchain in IoT applications. The objective of this research was to gain insight into algorithms predicting how IoT-DAG DLT horizontal scalability changes with increasing node count in a heterogeneous ecosystem of full and light nodes. It specifically questioned how incorporating preferential attachment topology impacts IoT network scalability and performance, focusing on transaction throughput and energy efficiency. Using an Agent-Based Modelling (ABM) simulation, the study evaluated a heterogeneous 1:10 full/light node network with Barabási Albert Preferential Attachment (PA-2.3) across increasing node counts (100-6400). Performance was measured by Confirmed Transactions Per Second (CTPS) and Mean Transaction Latency (MTL). Results showed CTPS scales linearly with node count ($R^2 \approx 1.000$), exhibiting robust predictability. MTL increased logarithmically ($R^2 \approx 0.970$), becoming more predictable as the network grew. Horizontal scalability showed exponential decay. The study confirms that IoT-DAG DLTs with preferential attachment can achieve predictable, near-linear throughput horizontal scalability, highlighting that topology matters and optimising CTPS yields the highest throughput gains.

KEYWORDS

Direct Acyclic Distributed Ledger, Horizontal Scalability, Internet of Things, NetLogo, Model Simulation, Preferential Attachment Topology

1. INTRODUCTION

Directed Acyclic Graph (DAG) based Distributed Ledger Technologies (DLT) have been gaining attention for their potential to address the scalability and energy efficiency challenges faced by traditional blockchain-based systems in Internet of Things (IoT) applications[1]. Unlike traditional blockchain, e.g. Bitcoin [2], [3], which organises transactions into blocks, DAG structures transactions as vertices and edges, allowing for parallel processing and thus higher throughput. This makes DAG-based systems like IOTA Tangle[4]particularly suitable for IoT environments where numerous devices need to communicate frequently, securely and efficiently.
The use of agent-based computational models is an emerging tool for empirical research to study behaviour among "bottom-up" models [2]. In agent-based modelling (ABM), each agent follows a set of rules and behaviours, and these agents collectively form a dynamic system that can help researchers gain insights into the emergent properties of complex systems. Agent-Based Modelling (ABM) tools are discrete event simulators which have shown increased popularity in the simulation of theoretical models such as virus propagation in [5], which is similar to message

DOI: 10.5121/ijnsa.2025.17304

propagation in distributed systems. ABMs have been used recently in simulating IoT devices, as shown in recent studies by [3] and [4]. Large networked systems, such as the Internet of Things integrated with Distributed Ledger Technology (IoT-DLT), are a challenge when performing empirical experiments, due to the number of physical devices and the associated costs. ABM tools can be used to simulate an IoT-DLT environment and perform experiments on emergent behaviour while providing significant savings in material costs and time. The common goal of most generalised ABM simulators is to provide a layer of abstraction and permit modellers to focus on the development of agent-based models rather than on their implementation.

The key contribution of this research is to gain some insight into the algorithms that can predict how an IoT-DAG DLT scalability changes with increasing node count of a heterogeneous ecosystem of full and light nodes. The key research question of this research is

- How does the incorporation of preferential attachment topology in a DAG-based DLT impact the scalability and performance of IoT networks, particularly in terms of transaction throughput and mean transaction latency for both full and light nodes?

The rest of this paper is organised as follows: Section 2, related studies, Section 3, the methodology used, Section 4, the results of the performance test, Section 5, a discussion analysing the results and Section 6, the conclusion of the research. The IoT-DAG-DLTSim Model is part of an ongoing study on improving the scalability of the IoT-DLT ecosystem.

2. RELATED STUDIES

2.1. Horizontal Scalability of Distributed Systems

Network Topology Studies in Distributed Systems, Preferential Attachment Topology: Barabási-Albert Preferential Attachment (PA)[6] topology has been widely studied for its ability to model real-world networks with scale-free properties. The PA topology is characterised by a power-law degree distribution, which can be useful in studying the effects of node increases in distributed systems. Research has shown that preferential attachment can lead to more robust and scalable networks by creating a few highly connected nodes (hubs) that can efficiently distribute information. This topology can be particularly relevant in the context of IoT distributed ledgers, where scalability is a critical concern. Previous simulation studies have focused on evaluating the performance of different consensus algorithms and network topologies under varying conditions. For example, simulations have been used to assess the impact of node increases on network latency, throughput, and energy consumption. These studies provide valuable insights into the scalability challenges and potential solutions for distributed ledger technologies.

Research by[1]explored lightweight and scalable blockchain solutions optimised for IoT requirements. For example, the Lightweight Scalable Blockchain (LSB) aims to enhance scalability for IoT by reducing computational overhead. Similarly, the concept of using management hub nodes or high-resource devices to handle communications has been proposed to overcome the limitations of traditional blockchain in IoT settings. These approaches focus on minimising energy consumption and improving transaction-processing speed, which are critical for resource-constrained IoT devices.

In the review by [7] on the technical and security issues of integrating IoT and distributed ledgers, the authors identified the problem of scalability, which is characterised by the need to support an increasing load of transactions, in addition to the increasing number of nodes within the network.

2.2. Scalability Performance Measurement

According to [8] the transaction processing time equation of a distributed system can be summarised as in Eq. (1)

$$T = t_i + t_c = (t_v + t_{pow} + t_{nx} + t_e) + t_c \quad (1)$$

where t_i is the issuance time, $t_c$is the confirmation time $t_v$is the validation time $t_{pow}$is the PoW time, t_{nx} is the network overhead, e.g. encryption/decryption, hashing and authentication. In [9], a study comparing various DLT designs, the authors identify three performance characteristics and their operationalisation shown inTable 1

Table 1: DLT Performance characteristics and measurement ([9])

DLT Performance Characteristic	**Definition**	**Operationalisation**
Confirmation Latency (P1)	The average time until enough blocks (or transactions) are added to the distributed ledger so as to reduce the likelihood of tampering a previously added block or transaction is below a certain threshold.	$SecurityConfirmations * BlockCreationInterval$ = mean transaction latency (MTL)
Throughput (P2)	The number of transactions validated and appended to the distributed ledger in a given time interval.	$completedtransactionstime/ interval$ – Confirmed Transactions Per Second(CTPS)
Scalability (P3)	The ability of a DLT design to handle an increasing amount of workload or its potential to be enlarged to accommodate an increasing number of nodes	$\dfrac{\left(\frac{\boldsymbol{Throughput\ (k2)}}{\boldsymbol{MTL\ (k2)}}\right)}{\left(\frac{\boldsymbol{Throughput\ (k1)}}{\boldsymbol{MTL\ (k1)}}\right)}$ $\boldsymbol{where\ MTL, CTPS > 0, k1 < k2}$

Adapting the formula by [9] the scalability can be calculated by using the formula Eq. (1)to derive a formula for calculating scalability due to a change in the number of nodes, i.e. from k_i to $k_{i+\Delta i}$

$$\Psi = \frac{\varphi(k2)}{\varphi(k1)} = \frac{\frac{CTPS(k2)}{MTL(k2)}}{\frac{CTPS(k1)}{MTL(k1)}} = \frac{CTPS(k2) * MTL(k1)}{CTPS(k1) * MTL(k2)} \quad (2)$$

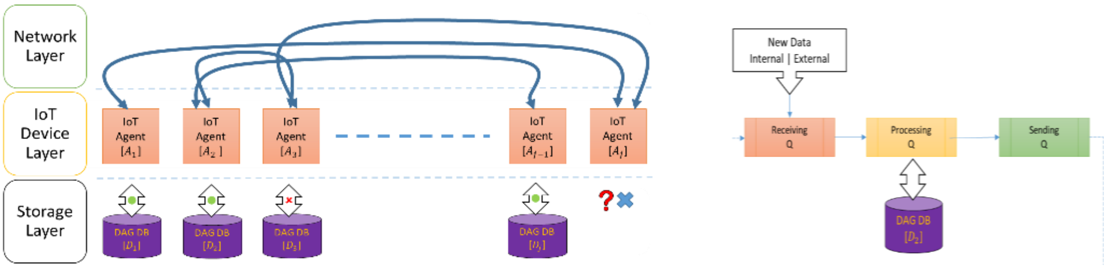
,where the function$\varphi(k_{i=[1..N]})$ is the ratio at a given time at node one of the nodes $i = \in 1..N$. If the value of the Ψ is one(1), then scalability remains constant. If the Ψ reduces below one(1), then scalability has reduced; if more than one(1), then scalability has increased.

While most studies focus on improving census protocols, energy efficiency, few examine the role light nodes play in IOT-DAG networks at scale. This research focused on studying the effect of node increase, with a heterogeneous mix of light and full nodes, using a simulation. This allows us to derive predictive metrics on the behaviour of the IoT-DAG-based DLT ecosystem. Though it explores using a laboratory environment, it nevertheless can provide insight into how networks behave as the number of nodes increases.

3. METHODS AND MATERIALS

3.1. Model Experimental Design and Setup

In the experiment, the performance of the IoT-DLT model was measured in terms of CTPS and MTL, and the information used to calculate the simulation model's scalability using the formulas identified in (1) and (2). The **Error! Reference source not found.** shows a high-level design of the IoT-DAG-DLT ecosystem. The **Error! Reference source not found.**(a) shows a complete ecosystem with a set of Agents $A \in [A_1, A_2 \dots A_j]$, some with active ledgers while others are available/unavailable at specific times. Each agent can connect to other agents via the network area. The **Error! Reference source not found.**(b) shows the design of individual agents, which use a gossip-like protocol to receive messages, process them and send them to other agents.



(a) (b)

Figure 1.(a) DAG-IoT-DLT Ecosystem Design, Figure 1.(b) Transaction processing design

During each epoch, each Agent randomly produced transactions with a probability of 0.01, while full-agent with a data store processes the transactions received or generated by updating the local database. A Gossip SIR[10] protocol was used to propagate the transactions to the neighbouring agents. New transactions are appended to the DAG database as new tips by attaching them to two new existing tips in the local database, after which they are Gossiped using the SIR [10]algorithm to the neighbouring agents for replication. When transactions are received for replication, the DAG ledger attaches them to the corresponding DAG tips if they have been received, otherwise, the DAG Ledger waits for the required tips. If it takes too long to get the required DAG transactions, the agents request them from the neighbouring agents using a Gossip SIR algorithm. The key components of the agent model are:

- **Agents**: Full-agents with a local DAG-Database, a light-agent with no DAG-Database
- **Agent network**: Agents are interconnected using a scale-free random network using Barabási-Albert Preferential Attachment (PA) [11] algorithm, which simulates internet-like connections with k-degree of 2.3 as per empirical data from Barabási and Albert [6] on statistical mechanics of complex networks.
- **DAG Ledger**: Each agent maintains a local database to store new transactions and replicate transactions of neighbouring agents. The DAG Ledger is built using an IOTA [12] like random algorithm.
- **Gossip Protocol**: Agents send and receive messages using the random Gossip SIR model [10], which performs a push for new messages and a pull for any missing DAG tips based on the age of the updates.

The **Error! Reference source not found.** shows the design of the IoT-DAG-DLTSim model, which simulates a decentralised ledger system using a directed acyclic graph (DAG) architecture within a scale-free network topology. Initialisation begins by defining critical parameters: network size (ranging from 100 to 6400 nodes), a 1:10 ratio of full to light nodes, and a power-law topology ($\gamma = 2.3$) to emulate real-world peer-to-peer networks. Each simulation epoch executes a transaction lifecycle where nodes probabilistically generate transactions, which are validated and appended to the DAG as new tips. Transactions may be received from neighbouring nodes, and the receiving agent checks if they are already attached and appends them to the DAG ledger or discards them if duplicates are detected. Valid transaction as sent to neighbouring nodes through a gossip-based propagation protocol. The model captures global state snapshots at each epoch, and at the end of the experiment, extracts performance metrics including Mean Transaction Latency (MTL), Confirmed Transactions Per Second (CTPS). To ensure statistical reliability, the simulation iterates 30 times per parameter set, with outputs exported to structured datasets for analysis. This design enables the study of emergent behaviours in heterogeneous networks while quantifying trade-offs between latency, throughput, and network growth dynamics under varying conditions.

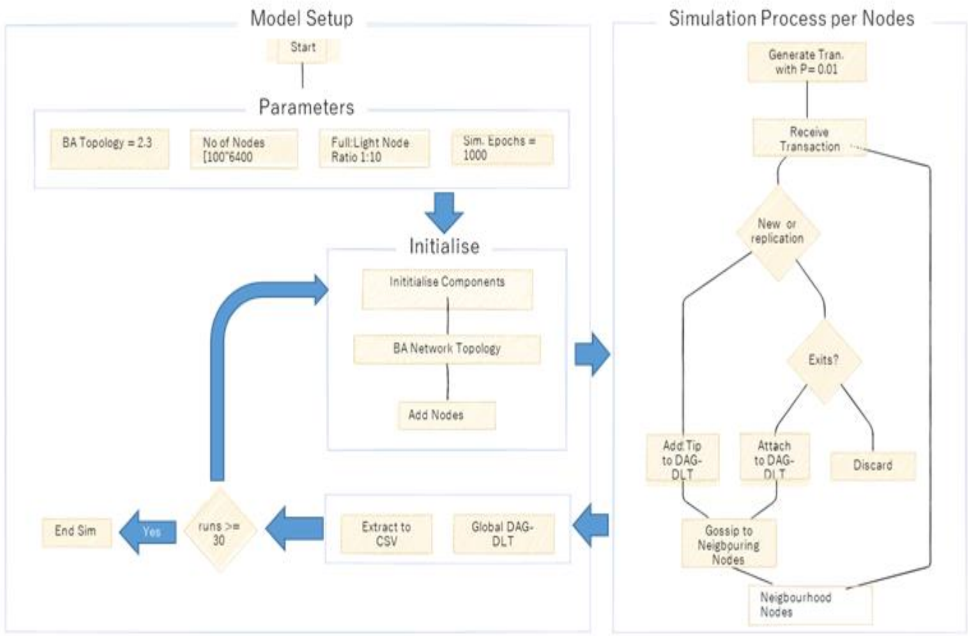


Figure 2. Model design flow diagram, showing simulation flow with setup and simulation process per node

The descriptive flow diagram in Mermaid[13] code format is shown in Figure 3, There are two key parts: the simulation model setup used to select the parameters used to run the simulationand select the number of iterations and the run part, which executes the simulation and collects the data.

Simulation Model Work Flows Descriptions

```
flowchart TD
 subgraph S["Model Simulation"]
      S1["START"]
      A["A"]
      B["B"]
      D["D"]
      S2{"simulations > 30"}
      E["E"]
 end
 subgraph A["Parameters"]
      A1["Number of Nodes: 100, 200, 400, 800..."]
      A2["Full:Light Node Ratio = 1:10"]
      A3["PA Topology Parameter = 2.3"]
      A4["Simulation Epochs = 1000"]
 end
 subgraph B["Initialization"]
      B1["Initialize model components"]
      B2["Generate PA Network Topology"]
      B3["Add Agents"]
 end
 subgraph C["Processing"]
      C1["Generate Transaction with probability P"]
      C2["Receive Transaction"]
      C3{"New or Replication?"}
      C4["Add Tip to DAG"]
      C5["Propagate to Neighbours"]
      C6{"Already in DAG-DLT?"}
      C7["Discard Transaction"]
      C8["Attach to DAG-DLT"]
      P1["Neighbouring Node"]
 End
subgraph D["Data Collection"]
      D1["Global DAG-DLT View"]
      D2["Extract DAG_DLT -> CSV"]
 end
 subgraph E["End"]
      E1["End simulation"]
 end
S1 --> A
   A --> B
   B1 --> B2
   B2 --> B3
   B --> C
   C --> D
   D --> S2
   S2 -- Yes --> E
   S2 -- No --> B
   C1 --> C2
   P1 --> C2
   C2 --> C3
   C3 --> C4 & C6
   C6 -- No --> C8
   C8 --> C5
   C6 -- Yes --> C7
   C4 --> C5
   C5 --> P1
```

Figure 3. Mermaid pseudo code

Each full node stores its own transactions. To minimise memory and improve performance, a binary structure was used to store each transaction. An existing binary string extension was modified to accommodate this new requirement.A data structure was designed to store the global view of all the full nodes, to allow capture of confirmed transactions per second and the mean transaction latency.

3.2. PA Model Configuration

APA of 2.3 seven (7) network topology configurations were created for 10, 20, 40, 80, 160, 320 and 640 full modes. Then 90, 180, 360, 720, 1440, 2880 and 5760 light nodes respectively were singularly connected randomly to each of the full nodes. A static configuration of the PA modelwas used to allow for reproducibility of results.

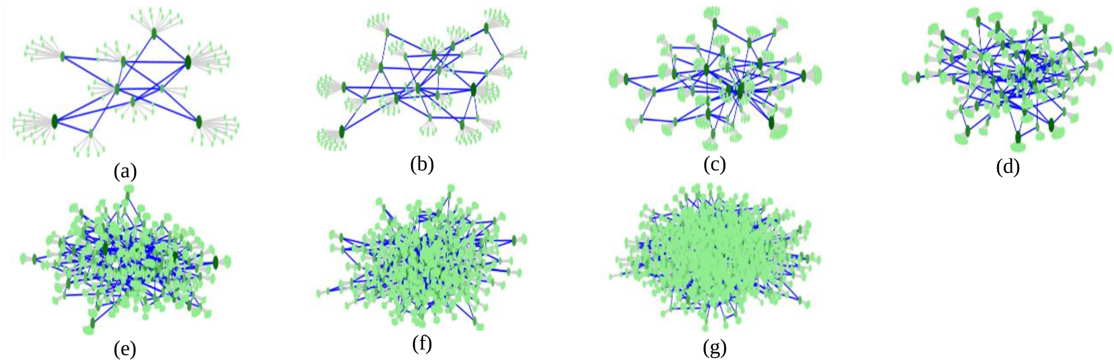


Figure 4. PA topologies k-degree =2.3 showing full and light agents connections, for seven (7) topologies [(a) = 100, (b) = 200, (c) = 400, (d) = 800, (e) = 1600, (f) = 3200, (g) = 6400 agents respectively

3.3. Model Parameters

The Table 2shows the parameters available and used to set up and run the experiment. The input parameters determine the size of the network and are changed during successive experiments using the NetLogo "Behaviour Space" Tool. The remaining inputs remain the same.

Table 2. Model input and output parameters

Parameter	**Values**
Input	
Number of agents in the DLT-DAG network (N_{total})	[100, 200, 400, 800, 1600, 3200, 6400]
Agent Types (N_{full}, N_{light})	[1 ,10]
Agent Ratio	$N_{light} = 0.1 * N_{total}, \quad N_{full} = 0.9 * N_{total}$
Mean degree of agent connectivity ($k - degree, \kappa$)	2.3
Agent Heterogeneity (H)	0
Epochs per experiment (i)	$i \in [1 \ldots 1000]$
Transaction generated per epoch (λ_i)	$0.01 * N_{full}$
Output	
Time taken to set up experiment(seconds)	T_{setup}
Time taken to run experiment(seconds)	$T_{experiment}$
Transactions per epoch	λ_i
Transactions generated per experiment (Λ)	$\sum_{i=1}^{1000} \lambda_i$
Global Transaction Confirmation (β)	$\lambda(t_{confirm})_i, t_{confirm} \neq = -1, \sum_{j=0}^{N_{full}} t_{confirm} > 0.67$
Transaction Latency (δ)	$\delta = \lambda(t_{confirm} - t_{creation})_i$
Confirmed Transactions (i)	$\sum_{i=1}^{n} \lambda_i, t_{i+n} \neq -1$

3.4. Equipment Configuration

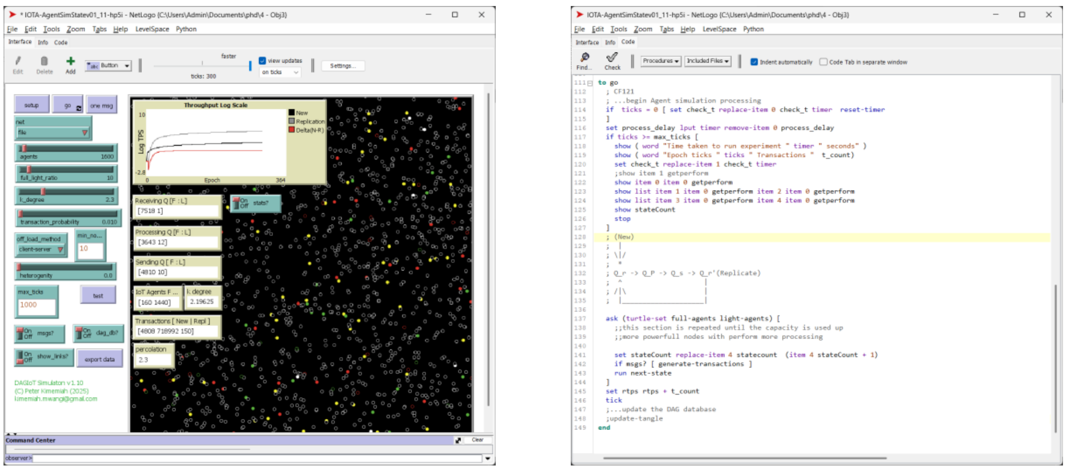
The model was implemented using the popular NetLogo 6.3 ABM tool based on Java Virtual Machine (JVM), allowing operations on both Windows and Linux-based system. The Table 3 indicates the configurations of the equipment configuration.

Table 3. Experimental computing tools, setup and configuration

Configuration (Toshiba i3)
• Toshiba Satellite C850 • Intel, Core i3-2348M, 2-Core Processor, 4 logical processors 2.3GHZ, 12GB DDR3 RAM, L1 128KB, 512KB, 3MB • Intel HD Graphics 3000, • 400GB SATA HDD • Windows 10 pro, 10.0.19045,64bit • Java SE JDK 17.0.4 (64bit) • NetLogo 6.3, extensions (array, bitarray, csv, ls, lt, nw profiler, py, table, time) • Julia 1.10, Agents.jl 5.17

3.5. Model Execution

The simulation interface on NetLogo 6.3 is shown in
(a)along with the main execution code in
(b). NetLogo has a behaviour tool that allows for running multiple experimental runs, using various parameters unattended.



(a) (b)

Figure 5. Simulation screen Netlogo 6.3 configured for 1,600 nodes, with full:light node ratio of 1:10, PA of 2.3.

Each model was run 30 times for varying network sizes of N_{total}100, 200, 400, 800, 1600, 3200 and 6400 IoT agents. In this experiment, all other parameters were kept constant, i.e. $[N_{full}, N_{light}] = [1:10], \kappa = 2.3, \lambda = 0.01, \iota = [1,1000],\ Homogentity = 1$. A total of 30 runs x 7 agent configurations = 210 iterations were performed on NetLogo ABM on a Toshiba i3 running Windows 10 64-bit and 12 GB of RAM.

3.6. Model Dataset

The data was extracted from the Global DAG Ledger, and 840 records were collected for the experiment.

exhibits characteristics of data collected that included the CTPS and MTL calculated along with respective standard deviation, standard error, kurtoses, skewness and number of transactions over 1000 epochs.The data was gathered from the pseudo-DAG-database stored in a NetLogo [14] Table structure into CSV files, which are then uploaded into Julia 1.10. [15] DataFrames.jl in for analysis using Julia 1.10 Curvefit.jl, HypothesisTests.jl, GLM.jl, StatsBase.jl, StatsModels.jl libraries and GLMarkie.jl, StatsPlots.jl for producing the graphical outputs.

```
observer: "10:03:26.853 am 22-Apr-2025"[-1 1600 10 2.3]predefined PA network topology
observer: "Configuration : Agents 1600 ratio 10 % with [160 1440]"
observer: "            k-degree 2.3 transaction arrival 0.01"
observer: "Time taken to setup 0.01 seconds"
observer: "Time taken to run experiment 78.3 seconds"
observer: "Epoch ticks 1000 Transactions 15972"
observer: "10:03:26.852 am 22-Apr-2025"
observer: [["setup" 1600 10 2.3 0.01 0 "client-server"] [1000 15972 2492473 [35.352 78.304]]]
observer: [["CTPS" 15.89746192893401 6.773682007733509 0.215827454917899 0.4335050284159694 0.4107940642031047
985] ["MTL" 20.516444217382976 2.9865637248332293 0.023866557095145038 0.3371101251388722 0.4158100425554734
15659]]
```

Figure 6. Onscreen simulation model output for 1600 nodes and 1:10 full:light node ratio, with a PA of 2.3 and a transaction arrival rate probability of 0.01 over all nodes

4. RESULTS

The following section presents the results of the simulation analysis, starting with runtime details, descriptive summaries, and inferential insights. The results were also visualised using line graphs.

4.1. Simulation Runtime

At the onset, a comprehensive simulation was conducted across varying node counts to evaluate the impact on key performance metrics, including CTPS, MTL, and horizontal scalability. The experiment took a total of 32.79 hours with a standard error (SE) of 0.16 and a standard deviation (SD) of0.63 to run the complete sets of simulations as shown in Table 4.

Table 4. Full experiment runtimes in seconds(s) 30 runs per node count

Nodes	Experiment Runs	Toshibai3 Execution time(s)	Standard Error(±sε)	Standard Deviation(±σ)
100	30	55.02	7.61	15.29
200	30	127.41	4.43	20.14
400	30	344.55	4.09	13.24
800	30	1,345.20	12.22	52.28
1600	30	5,469.90	40.45	183.37
3200	30	22,040.25	117.20	537.76
6400	30	88,669.23	506.05	2,204.76
	Total Seconds	118,051.56	578.28	2,277.51
	Total Hours	32.79	±0.16	±0.63

Each simulation was run 30times in order to obtain a mean average due to the stochastic behaviour of the IoT Devices. Both SE and SD increase with larger node counts (e.g., from 100 to 6400 nodes), which is expected because runtime grows exponentially (e.g., 1.591s →

2946.44s), so variability scales proportionally. This adequately reflects horizontal scaling.The relative error (SE/μ) remains small (e.g., for 6400 nodes: SE/μ = 14.138/2946.44 ≈ 0.48%), suggesting consistent precision despite larger runtimes.

4.2. Descriptive Analysis of CTPS and MTL

This section provides a descriptive analysis of the CTPS and MTL behaviour of the IoT-DAG model based on data collected over 30 experiment runs. The Table 5summarises the descriptive statistics for CTPS and MTL collected over 30experiment runs, for a PA of 2.3 for full nodes and a client-server connection for light nodes, with a full-to-light nodes ratio of 1:10 and a transaction probability of 0.01 for each node. The symbol μ is the mean, the σ is the standard deviation, and sε is the standard error calculated for each experiment set. The transactions count represents the mean number of transactions processed by each full node.

The data in Table 5demonstrates strong linear scalability with increasing network size (nodes), as evidenced by:

- Near-perfect doubling of transaction throughput (Count) as nodes increased from 100 to 6,400 (998.8 → 64,046.1), validating the efficiency of the BA-2.3 topology.
- CTPS scaled linearly (μ: 2.456 → 63.548), with tight confidence intervals (low *σε/σ* ratios < 0.005), indicating robust predictability.
- The stability of *σε/σ* across node sizes (e.g., 0.0024–0.0049 for CTPS) suggests the model's performance variability is independent of network scale—a critical feature for IoT deployments.
- On Latency (MTL) behaviour, MTL increased logarithmically with node count (μ: 12.222 → 22.638), reflecting the expected trade-off between network size and propagation delay.
- Declining *σε/σ* ratios (0.0025 → 0.0001) show that latency becomes more predictable as the network grows, likely due to the PA model's hub-dominated routing.

Table 5. IoT-DAG DLT Simulation Model: Descriptive statistics for CTPS and MTL for 30 runs per node, PA = 2.3, transaction probability 0.01/node, full:light ratioof 1:10

Nodes	Transactions		CTPS				MTL			
	Count	**sε**	**μ**	**sε**	**σ**	**sε/σ**	**μ**	**sε**	**σ**	**sε/σ**
100	998.800	5.317	2.456	0.0907	18.609	0.0049	12.222	0.087	34.859	0.0025
200	2,001.967	7.435	2.852	0.0749	23.609	0.0032	15.125	0.062	43.162	0.0014
400	3,985.133	15.000	4.473	0.1025	38.603	0.0027	16.658	0.046	53.709	0.0009
800	8,006.267	16.643	8.086	0.1492	60.718	0.0025	19.024	0.033	64.842	0.0005
1,600	15,992.733	23.698	15.904	0.2253	92.661	0.0024	20.553	0.024	78.414	0.0003
3,200	31,947.767	29.644	31.746	0.4113	169.196	0.0024	21.371	0.018	98.114	0.0002
6,400	64,046.133	51.342	63.548	0.5935	244.177	0.0024	22.638	0.012	114.439	0.0001

The Figure 7 shows the relative errors of the transaction CTPS and MTL data over the node count. The practical implications are that though larger networks introduce higher latency, the diminishing variability (*σε/σ*) suggests stable performance bounds for IoT applications

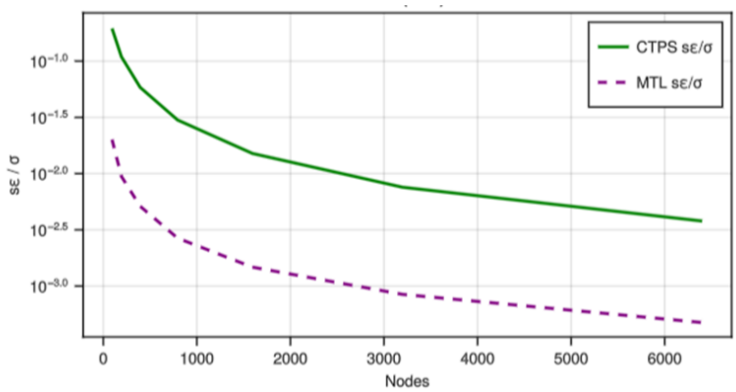


Figure 7. Relative errors of the transaction CTPS and MTL data over the horizontal scalability overvarying number of IoT nodes

The models derived show robustness as indicated by:

- Low standard errors (σε) for CTPS and MTL (e.g., CTPS σε:0.0907 → 0.5935) confirm the simulation's reliability across 30 runs.
- Consistent *σε/σ* ratios (e.g., ~0.0024 for CTPS at *n* ≥ 800) imply the model's stochastic elements (e.g., transaction probability 0.01/node) do not disproportionately affect outcomes at scale.
- Linear CTPS scaling in Figure 8 assumes perfect resource allocation; future studies should incorporate network and compute limits.

4.3. Model Fitting

Table 6 presents the relative standard error (RSE) percentages, model estimates, and model fit (R^2) for different parameters in the analysis. The Figure 8 plots the values of Table 6 and calculates the range of the SE percentage, the Model Estimate and the type of model and R^2 fit for each variable,

Table 6. Relative SE, model estimation and model fits R^2

Parameter	Relative SE % Range	Model Estimate	Model Fit
Transactions	$0.080\% \leq s\varepsilon \leq 0.532\%$	—	—
CTPS	$0.243\% \leq s\varepsilon \leq 0.487\%$	$CTPS \approx 0.699 + 10^{-2} * nodes$	Linear, ≈ 1.000
MTL	$0.010\% \leq s\varepsilon \leq 0.250\%$	$MTL \approx 15.882 + 10^{-3} * ln(nodes)$	Logarithmic, ≈ 0.970
Horizontal Scalability (ψ)	—	$\psi \approx 52.723 + e^{(-0.796 * nodes)}$	Exponential, ≈ 0.926

For the transactions, the RSE range is very low (0.080% to 0.532%), indicating high precision in the measurements. On CTPS (Confirmed Transactions Per Second), the RSE range (0.243% to 0.487%) remains low, reinforcing model reliability.

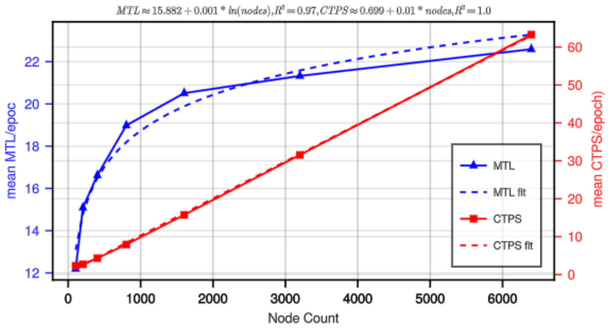


Figure 8. Visual representation of the model and curve fit for mean CTPS,MTL and their corresponding R2 values for transaction arrival of probability 0.01 per node full:light ratio of 1:10 for 30 runs per node count.

The linear model (CTPS $\approx$ 0.699 + 10^{-2} $\times$ nodes) fits the data almost perfectly ($R^2 \approx 1.000$), implying a strong direct relationship between CTPS and node count. The MTL, the RSE is extremely low (0.010% to 0.250%), indicating very precise estimates.In Figure 8 the logarithmic model (MTL $\approx$ 15.882 + 10^{-3} $\times$ ln(nodes)) shows a good fit of $R^2 \approx 0.970$, confirming that MTL increases logarithmically with node count, consistent with diminishing returns at scale. Scalability Estimation.

As indicated in Table 7, all predictors—network size (nodes), CTPS, and MTL, had statistically significant effects on transaction throughput (*p* < 0.001). The model explained [R^2]% of the variance in throughput, indicating strong predictive power. Each additional node increased throughput by ~2.33 transactions, supporting the scalability of preferential attachment networks in IoT DAG-DLTs. This aligns with Barabási-Albert (PA) topology principles, where larger networks sustain higher activity. Throughput was most sensitive to CTPS with every unit increase in transactions per second boosted throughput by ~766 transactions. This underscores the critical role of processing speed in IoT ledger performance. Not surprisingly, higher MTL correlated with increased throughput. This may reflect network buffering effects, where delays allow more transactions to accumulate before validation. However, this warrants further investigation to rule out confounding factors.

Table 7. Linear regression results predicting transaction throughput (trans) from network nodes, CTPS, and MTL

Dependent Variable: trans ~ 1 + nodes + CTPS + MTL

Variable	Coef.	Std. Error	t-value	Pr(>|t|)	95% CI
(Intercept)	-2832.48 ***	112.309	-25.22	<0.001	[-3053.91, -2611.06]
nodes	2.33 ***	0.283	8.25	<0.001	[1.77, 2.89]
CTPS	765.79 ***	28.198	27.16	<0.001	[710.19, 821.38]
MTL	143.89 ***	5.893	24.41	<0.001	[132.27, 155.51]

Signif. codes: *** p < 0.001

Scalability (ψ) Measurement over node growth using CTPS/MTL ratio is shown in along with the curve fit and its respective R^2. Horizontal Scalability (ψ) shows an exponential decay model (ψ $\approx$ 52.723 + $e^{-0.796}$ $\times$ nodes) fits reasonably well ($R^2 \approx 0.926$), implying that scalability benefits diminish rapidly as nodes increase.

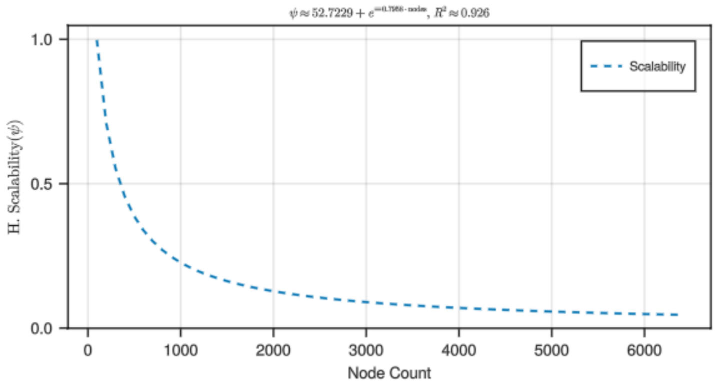


Figure 9. Analysis of effects of nodes on CTPS and MTL with curve fit and R^2 value

Estimation of the scalability of the IoT-DAG model through linear regression analysis was conducted. This evaluated the relationship between transaction throughput, network nodes, transactions per second (CTPS), and message latency (MTL), providing insights into the factors influencing performance in large-scale IoT deployments. The regression results highlight the significant predictive power of these variables, offering a deeper understanding of how throughput is impacted by network size, processing speed, and latency in the system as indicated in Table 7 and Figure. 9

For IoT applications, these results suggest:

- Topology matters: PA networks scale well with node growth.
- Hardware/software improvements to transaction processing will yield the higher throughput gains.
- Controlled delays might improve throughput but could compromise real-time performance.

5. CONCLUSION

The study confirms that IoT-DAG DLTs with preferential attachment (BA-2.3) and a 1:10 full/light node ratio can achieve predictable, near-linear scalability in throughput (CTPS) with modest latency growth. However, the model assumes ideal conditions; real-world deployments must account for physical constraints not captured here.

This research has demonstrated that, for the simulation models, CTPS scales linearly with node count, while MTL follows a logarithmic trend, indicating that adding more nodes yields diminishing gains in transaction load capacity. On horizontal scalability (ψ), the results exhibit exponential decay, which supports that initial node additions improve performance significantly, while the marginal benefit decreases sharply at higher node counts.

The high R^2 values and low RSE ranges suggest strong model reliability for CTPS and MTL, while the scalability model (ψ) provides useful but slightly less precise predictions. These findings can guide system design decisions, particularly in optimising node deployment for performance and scalability trade-offs.

REFERENCES

[1] A. Cullen, P. Ferraro, C. King, and R. Shorten, 'On the Resilience of DAG-Based Distributed Ledgers in IoT Applications', *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7112–7122, Aug. 2020, doi: 10.1109/JIOT.2020.2983401.

[2] H. Hellani, L. Sliman, A. E. Samhat, and E. Exposito, 'Tangle the Blockchain:Towards Connecting Blockchain and DAG', in *2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Bayonne, France: IEEE, Oct. 2021, pp. 63–68. doi: 10.1109/WETICE53228.2021.00023.

[3] S. A. Bragadeesh, S. M. Narendran, and A. Umamakeswari, 'Securing the Internet of Things Using Blockchain', in *Essential Enterprise Blockchain Concepts and Applications*, 1st ed., K. Saini, P. Chelliah, and D. Saini, Eds., Auerbach Publications, 2021, pp. 103–122. doi: 10.1201/9781003097990-6.

[4] J. Rosenberger, F. Rauterberg, and D. Schramm, 'Performance study on IOTA Chrysalis and Coordicide in the Industrial Internet of Things', in *2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, Dubai, United Arab Emirates: IEEE, Dec. 2021, pp. 88–93. doi: 10.1109/GCAIoT53516.2021.9692985.

[5] A. Mohammed, H. A. Jamil, S. Mohd Nor, and M. NadzirMarsono, 'Malware Risk Analysis on the Campus Network with Bayesian Belief Network', *Int. J. Netw. Secur. Its Appl.*, vol. 5, no. 4, pp. 115–128, Jul. 2013, doi: 10.5121/ijnsa.2013.5409.

[6] R. Albert and A.-L. Barabási, 'Statistical mechanics of complex networks'', *Rev Mod Phys*, vol. 74, no. 1, pp. 47-97, Jan. 2002, doi: 10.1103/RevModPhys.74.47.

[7] A. Cecilia Eberendu and T. Ifeanyi Chinebu, 'Can Blockchain be a Solution to IoT Technical and Security Issues', *Int. J. Netw. Secur. Its Appl.*, vol. 13, no. 6, pp. 123–132, Nov. 2021, doi: 10.5121/ijnsa.2021.13609.

[8] S. Park, S. Oh, and H. Kim, 'Performance Analysis of DAG-Based Cryptocurrency', in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, pp. 1–6. doi: 10.1109/ICCW.2019.8756973.

[9] F. Gräbe, N. Kannengießer, S. Lins, and A. Sunyaev, 'Do Not Be Fooled: Toward a Holistic Comparison of Distributed Ledger Technology Designs', presented at the Hawaii International Conference on System Sciences, 2020. doi: 10.24251/HICSS.2020.770.

[10] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, 'Using Gossips to Spread Information: Theory and Evidence from a Randomized Controlled Trial', *ArXiv14062293 Phys.*, May 2017, Accessed: Mar. 14, 2022. [Online]. Available: http://arxiv.org/abs/1406.2293

[11] A.-L. Barabási, R. Albert, and H. Jeong, 'Scale-free characteristics of random networks: the topology of the world-wide web', *Phys. Stat. Mech. Its Appl.*, vol. 281, no. 1–4, pp. 69–77, Jun. 2000, doi: 10.1016/S0378-4371(00)00018-2.

[12] C. Fan, 'Performance Analysis and Design of an IoT-Friendly DAG-based Distributed Ledger System', Masters, University of Alberta, Department of Electrical and Computer Engineering, 2019. doi: 10.7939/r3-21yj-3545.

[13] K. Sveidqvist and contributors, *Mermaid.js: Markdown-inspired diagramming tool*. (2020). [Online]. Available: https://mermaid-js.github.io

[14] Q. A. Chaudhry, 'An introduction to agent-based modeling modeling natural, social, and engineered complex systems with NetLogo: a review', *Complex Adapt. Syst. Model.*, vol. 4, no. 1, pp. 11, s40294-016-0027–6, Dec. 2016, doi: 10.1186/s40294-016-0027-6.

[15] M. Bouchet-Valat and B. Kamiński, '**DataFrames.jl** : Flexible and Fast Tabular Data in *Julia*', *J. Stat. Softw.*, vol. 107, no. 4, 2023, doi: 10.18637/jss.v107.i04.

AUTHORS

Mr. Peter Kimemiah Mwangi is presently a postgraduate PhD. research student and an assistant lecturer in the Information Technology department at the Murang'a University of Technology, Main Campus, Murang'a County, Kenya. He has 12 years' experience as an assistant lecturer and 22 years working in various senior positions in the IT Industry. He received his Master's in Information Systems from the University of Phoenix, Arizona, United States of America and a BEd. Mathematics and Physics from Kenyatta University, Nairobi, Kenya. His research interests include ERPs databases, the Internet of Things, computing infrastructure and networks.

Dr. Stephen T. Njenga is presently working as a Computer Science faculty member at the Murang'a University of Technology, Main Campus, Murang'a County, Kenya. He has 11 years of experience as a faculty member. He received his PhD in Information Systems from The University of Nairobi, Kenya. He received a Master's of Computer Science from The University of Nairobi, and a B.Sc. in Computer Science from Egerton University. His key research interest includes artificial intelligence, agent-based modelling and distributed ledger technology.

Dr. Gabriel Ndung'u Kamau is a Senior Lecturer at Murang'a University of Technology, with a Bachelor of education in Mathematics and Business from Kenyatta University, an MBA (MIS), and a PhD in Business Administration (Strategic Information Systems) from the University of Nairobi. He has over 10 years of teaching and administrative experience, including serving as Director of ODeL at his university. His research focuses on ICT4D, information systems philosophy, computer security, and disruptive technologies. Dr. Kamau has numerous publications, supervises PhD and Master's students to completion, and is a member of AIS and ACPK. He has also received professional training in ISO auditing, network security, DAAD NMT-DIES and digital education