Performance Analysis and Design of an IoT-Friendly DAG-based Distributed Ledger System

by

Caixiang Fan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Distributed ledgers provide many advantages over centralized solutions in IoT projects including but not limited to improved security, transparency and fault tolerance. However, in order to leverage them at scale, their well-known limitations, i.e., scalability and performance, should be adequately 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 terms of scalability and performability.

In this thesis, we first conduct a comprehensive literature review on both distributed ledger technology applications in IoT and the performance evaluation of such decentralized systems. Then we present a detailed technical overview of IOTA, following a contractive review of different DAG-based distributed ledger technologies.

Next, we propose a scalable transactive smart homes infrastructure by leveraging IOTA protocol and following the separation of concerns (SOC) design principle. Based on the proposed solution, an experiment with 40 home nodes is conducted to prove the concept at large scale in a cloud environment. The results show that our solution provides a high transaction speed and scalability, as well as good performance on micropayment which is important in IoT initiatives. We conduct an analysis and discuss how the new system breaks out the Blockchain Trilemma, which claims that it is almost impossible for a

blockchain platform to simultaneously reach decentralization, scalability and security. Based on our findings on scalability and performance, we conclude that the proposed DAG-based distributed ledger is an effective solution for building an IoT infrastructure for smart communities, in which local residents can freely and securely transfer values.

Finally, we rigorously study the performance of the ledger to examine its applicability for IoT projects in which a high throughput is required. More specifically, we investigate the IOTA system to answer two key research questions: 1) what is the confirmation rate in the system given the design parameters and 2) what will be the optimal waiting time for a user to resend its previously submitted but not yet confirmed transaction to the ledger? In order to answer these vital questions, we perform real experimentation, simulation and analytical modeling. Our findings reveal the impact of arrival rate of transactions, consensus algorithm, randomness of the weighted random walk for tip selection and network delay on the confirmation rate. By decomposing the transaction confirmations in each graph layer, we build an analytical layered model. Thanks to the analytical modeling, we shed some light on the distribution of confirmation process, which is leveraged to calculate the optimal time for resending the unconfirmed transaction to the distributed ledger. Our performance model can be used by IoT project designers to perform what-if analysis and capacity planning in advance of the real deployments, with high level of accuracy.

Preface

The research of this thesis has been conducted in the Dependable Distributed System Lab (DDSL) led by Dr. Hamzeh Khazaei at the University of Alberta. This thesis is extended from two publications. Chapter 4 and Section 2.1 are mainly formed by the conference paper *"Towards A Scalable DAG-based Distributed Ledger for Smart Communities"*, accepted in the 5$^{\text{th}}$ IEEE World Forum on Internet of Things, 2019. Chapter 3, Chapter 5 and Section 2.2 are mainly from the paper *"Performance Analysis of DAG-based Distributed Ledgers: A Hybrid Modeling of IOTA"*, submitted to the IEEE Transactions on Services Computing, Special Issue on Blockchain-Based Services Computing.

In this thesis, my original work includes the system design and data analysis in Chapter 4 and Chapter 5, protocol theoretical analysis in Chapter 3 and concluding analysis, as well as the literature review in Chapter 2. Moreover, I am responsible for all the experiments such as experiment design and environment setting up, data collection, processing and visualization, and model validation. The technical parts including smart community's system architecture design, scalability analysis and performance analytical modeling are conducted by myself under the supervision of Dr. Hamzeh Khazaei and Dr. Petr Musilek, from the Electrical and Computer Engineering Department, and Dr. Yuxiang Chen from the Civil and Environmental Engineering Department of the University of Alberta.

For performance analysis, most system simulations are conducted by Sara Ghaemi, a master student in DDSL lab. We extended DAGsim [56] simulation engine to support end-to-end performance analysis for DAG-based distributed ledger systems.

Acknowledgements

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

As for myself, I would like to thank my supervisors Dr. Hamzeh khazaei, Dr. Yuxiang Chen and Dr. Petr Musilek for their professional direction and help. The support provided by the DDSL lab mates is also gratefully acknowledged. In particular, I would like to thank my dear wife for her encouragement and support for my research.

Contents

List of Tables

List of Figures

List of Symbols

- λ: transaction arrival rate
- λ_{M}: milestone arrival rate
- α: randomness factor for weighted random walk
- d: distance between agents, reflects network delay
- $agents$: full node amount in a simulated network
- D: distances matrix for all agents in simulation

Glossary of Terms

Approve The examination process to check if a transaction is valid, e.g. identity, address, value time, etc. It is used interchangeably with validate or refer to.

blockDAG Directed acyclic graph with blocks as the vertices. Block is composed of one or multiple transactions.

branchTransaction One of the selected tips

Bundle A group of one or multiple related transactions in IOTA, which acts as a container and forces an atomic operation for all included transactions.

Confirmation rate The confirmed transactions per second or throughput of a distributed ledger system.

Consensus The mechanisms or protocols that make sure all network participants (nodes) are synchronized with each other and agree on which transactions are legitimate, confirmed and can be added to the distributed ledger. This is the most important concept in distributed ledger world.

COO Coordinator, generates milestones to confirm transactions; the currently (April, 2019) running consensus in IOTA.

CTPS Confirmed transactions per second

DAG Directed acyclic graph, provides data structure for the Tangle, where the vertices represent transactions or blocks, and edges represent approvals.

DLT Distributed Ledger Technology includes blockchain and DAG-based distributed ledger such as IOTA.

DPoS Delegated proof of stake, a consensus example, used in BitShares.

IOTA The first open-source DAG-based distributed ledger, cryptocurrency or protocol.

IREA Indirect references extraction algorithm

MCMC Markov Chain Monto Carol, is the technique IOTA uses to calculate the probability for randomly walking in the DAG to select two tips. In this algorithm, each walk step does not depend on the previous one, but just follows a pre-set rule.

MWM Minimum Weight Magnitude, indicates Proof of Work difficulty level. This is an important parameter setting for client node when issuing transactions. The default and minimum MWM values in IOTA test net and main net are 9 and 14, respectively. The PoW difficulty increases by 3 times as the MWM value increases by 1.

PBFT Practical Byzantine Fault Tolerance, a consensus example, used in Hyperledger.

PoS Proof of stake, a consensus example, used in Ethereum.

PoW Proof of work, a consensus example, used in Bitcoin.

RWT Reattachment waiting time, the time spent for users between two attachments in IOTA transactions.

sn Refers to all seen confirmed transactions by a full node in the context of ZeroMQ listening socket

Tangle The IOTA Tangle is a stream of interlinked and individual transactions, which are distributed and stored across a decentralised network of participants.

Tip A transaction without any validation in the Tangle.

trunkTransaction The other of the selected tips

tx Transaction, or refers to all seen transactions by a full node in the context of ZeroMQ listening socket

txDAG Directed acyclic graph with transactions as the vertices

URTS Uniform random tip selection

UTXO Unspent transaction output, one type of cryptocurrency transaction models, refers to an output of a blockchain transaction that has not been spent.

Chapter 1

Introduction

1.1 Background

Internet of Things(IoT) is experiencing an exponential increase in terms of the connected devices. This is due to the ubiquitous connectivity, billions of IP addresses with IPv6 and rapid development of 5G. According to Gartner report, the number of connected devices is expected to be over 25 billion by 2020 [41]. However, this tremendous market growth raises new challenges such as security and privacy [12], scalability and data processing performance for IoT system architecture, which means that an effective solutions need to be devised.

Distributed Ledger Technologies (DLTs), with the features such as decentralization, enhanced security and trust-free, obtained a lot of attention from both industry and academy to overcome the problems in IoT systems. Basically, there are two main types of DLTs according to different data structures for the ledger, which are block based Blockchain (BC) and blockless directed acyclic graph (DAG) based DLTs (e.g. IOTA Tangle [40]). DAG distributed ledgers can further be divided into $txDAG$ and $blockDAG$ according to their different data structures of vertices in the graph. BC is a distributed ledger for storing and sharing data across all nodes in a network. Based on different

usage contexts, various types of data including transaction record data (e.g. Bitcoin), contract and even personal healthcare information can be stored in a BC system. This emerging technology has drawn increased academic and industrial attention due to its attractive features including immutability, scalability and decentralization. According to a recent survey [5], there are about 42 industries (e.g. law enforcement, ride hailing and stock trading) that could be transformed by BC in the future. And this number keeps increasing, especially at the early stage of BC application innovation. Clearly, for many researchers and corporate CTOs, BC technology is potentially an effective solution to overcome the challenges in IoT [12].

Although BC has the potential to tackle the IoT problems such as security and privacy [12], its applicability to build an IoT architecture remains difficult. Firstly, the well-known limitations, i.e., scalability and performance, of standard BC systems make system designers hesitate. In addition, the application of BC in non-monetary IoT systems is not as straightforward as in electronic currency system such as Bitcoin [35]. There are many different types of DTLs, but no standards that would help identify the best one for IoT so far. For example, as for the participation permission, there are public, permissioned public/private and consortium networks; for transaction model, there are tokenized UTXO and non-tokenized account-based transactions [54]. Here, from the perspective of consensus, we list 5 types of main consensus mechanisms [16]:

- PoW (Proof of Work),
- PoS (Proof of Stake),
- DPoS (Delegated Proof of Stake),

- PBFT (Practical Byzantine Fault Tolerance), and
- Transaction References in DAG (Directed Acyclic Graph).

In BC based DLTs, the consensus mechanisms have many inherent disadvantages for IoT applications such as smart homes and communities. On one hand, the PoX consensuses are computationally expensive. According to the latest Bitcoin Energy Consumption Index, Bitcoin miners from all over the world consume over 70TWh of electricity every year to do the proof of work [11]. This is obviously not suitable for IoT scenarios with limited and light-weighted computation. On the other hand, the processed transactions per second (TPS) of the mainstream BC platforms like Bitcoin (7 TPS) and Ethereum (15 TPS) are very limited, because the single chain of blocks is linear, and blocks cannot be created simultaneously. For example, one Bitcoin block takes 10 minutes to be created and added to the main chain, which is very inefficient and fails to meet the requirement of instant transaction in IoT.

With DAG, transactions can be directly attached to a chain without waiting to be wrapped into a block in advance. More over, all new added transactions can be simultaneously run on different chains, which interwoven to form a network called Tangle [40]. Theoretically, the Tangle should be more efficient than traditional BC under the well-designed consensus mechanisms.

1.2 Motivation

From the perspective of Quality of Service (QoS), service level agreement (SLA) and BaaS (Blockchain as a Service), the transaction confirmation rate (throughput), average waiting time (transaction delay) and system scalability are extremely important for a DL system and user experience. Traditional BC system such as Bitcoin, Ethereum and Hyperledger has limited performance.

For example, Bitcoin will averagely take 10 minutes for a transaction to be confirmed once. And it usually recommends merchants to wait for 6 blocks' confirmations for a big transfer in the sake of safety, which means 1 hour to complete the payment process in a transaction. As a BC counterpart for IoT, the DAG-based IOTA claims to be scalable and to provide high performance, because of its innovative data structure and efficient consensus design for validation of transactions (TXs).

In this research work, we would like to explore the application effectiveness including scalability and performance of DAG-based DL IOTA in IoT scenario, e.g. the energy transactive smart communities. In particular, we focus on the performance of system throughput and transaction waiting time to reattach transactions for users, which we believe are critical for system designers.

1.3 Objectives

From an IoT system designer perspective, it is vital to know about the underlying system capacity in processing generated transactions in the network. Also, from the users/clients point of view, it is critical to know about the time that they need to wait before reattaching 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 decreased system efficiency. In this thesis, we aim to:

1. Design a decentralized IoT architecture for smart communities to conduct distributed electricity energy transactions. Develop a prototype on cloud to prove the concept by deploying a network of home nodes which are represented as virtual machines.

2. Explore the scalability of the proposed system. Through intensive transaction experiments, we aim to find more insights on the factors influencing scalability of IOTA.

3. Develop a hybrid performance modeling solution for evaluating the performance of DAG-based distributed ledger system. The models aim to accurately answer the performance questions such as confirmation rate (CTPS) and transaction waiting time to reattach etc.

More specifically, we strive to answer two vital questions about the performance of the system in a private IOTA network:

1. From the system design perspective, which factors influence the throughput of an IOTA system? And how, quantitatively, do they impact the throughput?

2. From the user perspective, what would be the optimal waiting time to wait for confirmation before reattaching the same original transaction?

1.4 Methodologies

To address above questions, we perform the following steps:

1. We study the system throughput by leveraging the DAGbased DL simulator for simulating IOTA to identify significant factors such as transaction arrival rate (λ), weighted tip selection randomness parameter (α), network delay reflected by distance (D) and different tip selection algorithms.

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 the first question.

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 good estimate for the second question.

1.5 Contributions

The contributions of this research work are as follows:

1. We propose a transactive smart communities IoT design based on IOTA. In this part of research, we argue that the DAG-based DL is an effective solution for designing a transactive smart homes architecture. The specific IOTA Tangle technology will be used to design our solution and conduct the experiments. In contrast to other IoT infrastructure proposals, our approach brings the following advantages to IoT architecture design:

    - **Scalability:** In a local community, the permissioned private network has a high scalability due to the decentralized DAG-based design and no transaction rate limit.
    - **Transaction speed:** The high transaction speed benefits from the DAG data structure and the efficient consensus mechanism. Transactions can be added to different "chains" in a Tangle simultaneously, which can speed up the transaction rate. For the largest TPS, an IOTA stress test held in April 2017 showed that the network had transaction processing capabilities of 112 Confirmed Transactions

per Second (CTPS) and 895 TPS within a small test network consisting of 250 nodes [55].

- **Security and privacy:** Our solution adopts IOTA Tangle which originally uses the hash function called Curl-p [40], then switches to the Keccak (SHA-3) for cryptographic signing. As for the 34% attack, we leverage the Membership Service Providers (MSP) from Hyperleger [21] as the authority management to set up a trustable environment, combined with the coordinators implementation to protect the ledger from 34% attack. Here, 34% attack refers to an attack such as double-spending by a group of participants controlling more than 33% of the network's computing power. Additionally, the private Tangle ensures all data are encrypted and stored in the locally running Home Nodes.
- **No transaction fees:** Tangle gets rid of "mining" using the following mechanism: before issuing a transaction, the node must confirm two previous transactions and do a very light-weighted proof of work. This means that all participants need to contribute their computation power to maintain the network to eliminate the transaction fees.
- **Micro-transaction:** Unlike Bitcoin with a threshold on the minimum amount of a payment, people can send as little as 1 IOTA in our solution, which is worth $0.572701 (as of September 10, 2018) and will always be available for sending without fee. This makes the M2M P2P micro-transaction possible for smart homes, such as in the case of energy transaction among neighbors in a local community.
- **Decentralization:** IOTA Tangle eliminates the notion of miners.

Every network participant only has access to limited computational resources. And anyone who wants to launch a transaction on the tangle needs to actively participate in the consensus. This makes our solution decentralized.

In summary, this research contributes to the design of a new scalable IoT architecture for smart homes and communities using DAG-based DLT. Our approach differs from other solutions in the way that it applies a lightweight, scalable and high-performance Tangle technology which is suitable for IoT.

2. We propose two performance models for IOTA Tangle, simulation model and layered model, which can provide the answer of question about the confirmation rate for DAG-based IOTA system in IoT situations. Compared to other performance analysis work on DAG-based DL systems, our work contributes on the following items:

   - **End-to-end performance analysis:** This research targets on an end-to-end question of throughput rather than attachment probabilities and tips number changing as the Tangle expands, which is more straightforward on performance evaluations and more beneficial to system designers.
   - **Simulator extension:** We make full use of DAGSim simulator for parameterized simulations, and extend it to support Coordinator consensus. This makes some performance related deep-level concepts, e.g. confirmations in a layer, visible and countable, which is impossible for real experimental environment.
   - **A hybrid performance model:** We build a hybrid model by combining the simulations and analytical modeling for confirma-

tion rate and confirmation distributions, in which we build a layered model by decomposing the simulations into DAG layers to look for confirmation distributions in graph layers. Thanks to this model, the reattachment waiting time of users is relatively accurately estimated.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 gives a brief overview of existing work on both DL applications in IoT and system performance evaluation including simulation and analytical modeling. Chapter 3 presents a detailed technical overview of IOTA. After that, we introduce a DAG-based DL system design for private transactive smart communities in Chapter 4, within which some basic experimental analyses are conducted to explore the system's scalability. In Chapter 5, we analyze IOTA system performance and propose the empirical and analytical models to answer the two previously mentioned research questions. Our experimental and simulation results, and main findings are also presented in this section. Chapter 6 discusses some topics on security and simulation model efficiency. Finally, Chapter 7 concludes the thesis and states some potential future directions of research.

Chapter 2

Related Work

In this part, we will review the related work from three perspectives. First, we introduce the research work on DL applications including smart homes communication network, IoT payment system and distributed IoT access control system. These works discuss a wide field of IoT-DL combination applications such as system designs, new consensus proposals and new protocol development. Second, we focus on the performance evaluation of proposed DL systems, both public and private, general and specific. Specifically, surveys on system simulation model and analytical model are conducted, respectively. Third, we review the existing DAG-based DL projects and current research work on this topic.

2.1 DLT Applications in IoT

A systematic literature review on the BC for the IoT was conducted in [10]. The survey explored whether the BC can be employed to foster a decentralized and private IoT by investigating factors that affect integrity, anonymity and adaptability of this technology. Similar to this survey, another work [7] took a deep look into how IoT and BC (especially smart contract) can be used together. The authors concluded that the combination of BC and IoT is powerful and can lead to significant changes across several industries, creating

opportunities for new business models and novel, decentralized applications [7]. Motivated by the positive conclusion, Novo proposed an IoT architecture for scalable access management in [38]. The decentralized access control system stored access control information using BC, which was developed to run as a single smart contract that defines the policy rules of the management system. However, unlike our solution, the previously mentioned systems had the limitations of transaction fees and processing speed from the inherited BC technologies [38].

In [14], the authors proposed a BC-based smart home architecture with a hierarchical structure consisting of three components: smart home, overlay network, and cloud storage. More specifically, [13] delved deeper and described the key components of smart home tier, in which an always online device played a role of "miner" to handle all transactions coming to or out of the smart home. This design provided an effective solution to overcome IoT security and privacy challenges by leveraging a new proposed BC called LSB (Lightweight Scalable Blockchain) [14], which adopted an IoT friendly consensus mechanism that eliminates the proof of work and incorporates a distributed trust method. In our solution, we share some features with the LSB such as no transaction fees, lightweight consensus mechanism and self-scaling. However, LSB employs the traditional BC which needs to wrap transactions into a block and wait for mining.

K. Yeow et al. [54] conducted a comprehensive review on decentralized consensus systems for IoT in terms of the data structure, consensus mechanism, and transaction models. From their proposed thematic taxonomy, a synthesized comparison between BC-based systems (e.g. Bitcoin and Ethereum) and DAG-based distributed ledgers (e.g. IOTA and Byteball) was conducted. By

analyzing and summarizing the pros and cons, the authors found that the DAG outperformed on scalability, transaction confirmation speed and decentralization. They concluded that DAG might be an answer to overcome the challenges of the fast scaling IoT with the need for low latency micro-payments in the machine-to-machine P2P decentralized infrastructure [54]. As an example, a new DLT-based charging and billing IoT architecture for electric autonomous vehicles (EAVs) was proposed in [47]. The authors leveraged IOTA based payment system through M2M communication (MQTT) to carry out micro-transactions for charging and billing in EAVs. In another research project, the authors utilized IOTA Tangle to present a streaming data payment protocol (SDPP) which was an application-layer protocol for enabling micropayments among IoT transactions [42].

Motivated by these innovative applications, we proposed a DAG-based IoT architecture for transactive smart homes leveraging IOTA Tangle in our previous work [15].

2.2 Performance Evaluation on DL Systems

According to Marsan [32], computer system performance evaluation can be conducted either through measurement or modeling. In this classification, shown in Figure 2.1, there are two main approaches to system modeling: simulation models, and analytical models such as Markov Chains [18] and Petri Nets [32].

2.2.1 Simulation Models

Simulation modeling is the process of creating and analyzing a simulator of a physical system to learn its behavior and predict its performance in the real world. Since DL-based systems are usually complicated and computational

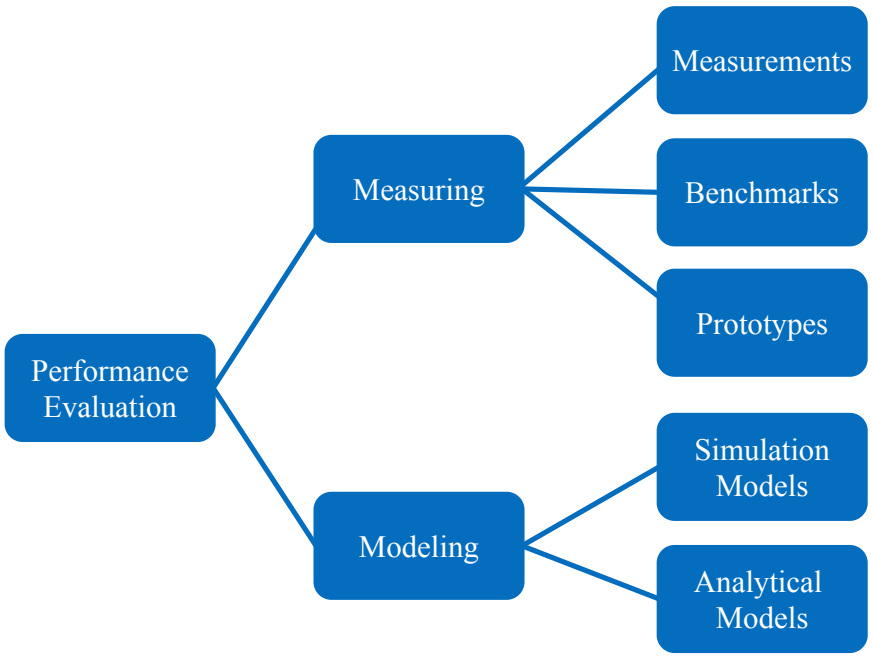

Figure 2.1: Performance evaluation and modeling

resource consuming, it is important to utilize simulation modeling to answer performance related questions such as throughput, latency, and scalability.

To explore the block creation performance under PoW consensus algorithm of BC, Alharby et al. [2] proposed BlockSim as a framework to simulate discrete-events in BC 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. So, it is hard to tell if the model is error-free and the simulator semantically works as intended. Similarly, in BC-based systems, Yasaweerasinghelage et al. [53] showed the feasibility of using architectural performance modeling 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 model to support architectural decision-making, especially on performance in BC-based system design.

Compared to BC systems, DAG-based DL systems are usually more complicated in terms of data structures and consensus achieving, and so more difficult to understand. Therefore, many modeling studies on DAG-based DL (especially IOTA) have been performed from different performance perspectives, providing good resources for learning about and designing DAG systems. For example, to help understand the Tangle, two IOTA Foundation white papers [3] and [25] 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 discussed the relationship between cumulative weights of transactions and tip numbers over discrete time steps. They found that the cumulative weight contains two phases of growth, namely the exponential and the linear. The simulation results also revealed that the numbers of tips $L(t)$ as a function of time remained stable under random tip selection strategy. In contrast, for Markov Chain Monte Carlo (MCMC, see Section 3.5 for details) guided tip selection strategy, it was stable only for small α (0.001) in examined time intervals.

The later paper [25] 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 [40]. 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 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 posi-

tion 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 performance metrics.

In order to illustrate and visualize the Tangle, Gal [17] developed an IOTA visualization simulator. Through this simulator, one could intuitively observe different tangles generated under various tip selection algorithms such as uniformed 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′shape and the transaction confirmation ratio. However, the total transaction number in this simulator was limited to 500 and performance metrics were not quantified.

To break out this limitation, Lathif et al. [26] proposed a configurable and interactive DAG-based DLT simulation framework named CIDDS [26] to enable large-scale simulations with thousands-nodes level. Moreover, Bottone et al. [4] presented and developed an extendable multi-agent simulator, in which the authors employed NetLogo [37] to provide a 3D visualization of the Tangle.

Similar to the previously mentioned BlockSim [2], Zander et al. [56] 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 simulation of the IOTA [40] protocol. In their work, the authors presented and implemented an asynchronous, continuous-time, and multi-agent simulator for DAG-based cryptocurrencies. By modeling both honest and semi-honest actors [56], 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 $\mathcal{O}(n^3)$ in the implementation. Therefore, it was very inefficient and not suitable for large-scale simulations, for instance, more than 10,000 transactions. Also, this research work did not go further into directly evaluating performance metrics rather than exploring the transaction attachment probabilities from each node. Our work borrows this simulator and builds into it the Coordinator confirmation consensus to develop a performance simulation model. Further more, we leverage the extended simulator to simulate IOTA for conducting performance study by setting different configurations.

2.2.2 Analytical Models

Analytical models leverage mathematical terminology, formulae and/or tools to describe the relationship between performance metrics and parameters in a computer system. They are very powerful to reveal the nature of modeled systems.

Sukhwani et al. [49] 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 parameter was conducted to examine the performance of larger networks. With extended empirical analyses on two releases of Hyperledger Fabric, H. Sukhwani concluded in [48], that the presented Stochastic models could be used to develop Fabric Network management infrastructure.

Li et al. [31] proposed a blockchain queuing theory of blockchain systems and provided system performance evaluation. Specifically, a Markovian batch-service queueing 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 system stable condition and analyzed three key performance metrics including mean transactions number in the queue, mean transactions 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 commonly used Markovian chain approach. As mentioned in their conclusion, this analytical model had the potential to open a series of promising research in queuing theory of blockchain systems, even though the proposed model was simple [31].

Motivated by this work, Saulo [44] 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′ law and standard M/G/1 queuing model [34], they found a simple relationship between E(B)-mean time between block confirmations, 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 transactions confirmation.

Recently, Memon et al. [33] proposed a queuing theory-based model by segregating the BC 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 BC 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*.

These related works reviewed above provide different perspectives to explore the system performance of DAG-based IOTA. However, before describing our models, we present a technical overview of IOTA protocol.

2.3 DAG-based DLTs

Different from the standard blockchain, many other distributed ledger systems chose DAG as the data structure. According to a recent comparative analysis conducted by H. Pervez et al. [39], there existed at least the following DAG-based distributed ledgers so far, IOTA, Byteball, Orumesh, Dagcoin, Nano and XDAG. In all these DLTs, without doubt, IOTA [40] took the most attention and became the leader in terms of both technology development and market capability. Established on its core revolutionary DAG-based distributed ledger

technology, the Tangle [40], IOTA proposed a fully decentralized peer-to-peer solution by requiring every participant to approve two previous unapproved transactions or tips, which were selected by a weighted MCMC random walk algorithm. This Tangle/DAG structure facilitated high scalability of transactions. The more participants in the Tangle, the quicker transactions could be confirmed [39].

In 2015, A. Churyumov [8] leveraged DAG to design a DL system named Byteball [8], in which a total order of transactions was established to protect itself from double-spends. This was achieved by selecting a chain (called main chain), which gravitated towards "units" issued by commonly recognized reputable users, i.e., 12 witnesses [8]. Therefore, Byteball separated transaction validators and issuers, and relied on trusted participants to get rid of PoW in reaching consensus, which led to transaction fees.

In the same year, S. Lerner [30] posted his draft (drafted already in 2012) of a coin based on DAG chain, which he named Dagcoin [30]. But, this never became a project and remained just a draft, so no actual coin was created until 2018. Dagcoin was initially built on top of the Byteball network by Y. Ribero and D. Raissar [43].

Motivated by IOTA and DAGCoin, J. Ahmed proposed OruMesh [1], which was a txDAG DL built on AMesh, a DAG-based highly decentralized data structure merging transactions and blocks and turning each transaction into a reward based processing operation. It utilized a social consensus-based algorithm to replace PoW for securing the network. Unlike doing PoW in IOTA, any participant was required to pay a computing power equal to the size of added data in Oru coin before adding a new tip to ledger. Similar to Byte-

ball, the transaction confirmation relied on selected witness (called testifier or OruPartners [1]). In order to reach stability, participants need to accumulate enough testifier-authored transactions on the AMesh after the newly added transaction. So, to minimize the confirmation period, the testifiers should post transactions frequently enough but not too frequently. And the best confirmation times were reached when the testifiers were well connected and run on fast machines so that they were able to quickly validate new transactions. The estimated best confirmation time was around 3 seconds [1].

Another main player was Nano [29], or called RaiBlocks [28], which utilized a novel DAG-based architecture called "block-lattice" [29] and achieved its consensus through a balance-weighted vote on conflicting transactions. This type of ledger recorded balance of an address through four types of transactions, i.e., *open*, *send*, *receive* and *change* [29], which were accordingly conducted by senders or receivers to complete a transaction. In Nano, all participants were required to do PoW similar to Hashcash before issuing any type of transactions. Even though the notations "transaction" and "block" were used interchangeably for Nano, it was still a block-less $txDAG$ rather than $blockDAG$ DL according to the definition of its "block" [29]. However, different from IOTA's vision of leading "machine-to-machine communication, commerce, data storage" [36] and becoming the premier protocol of IoT devices, Nano focuses on "reliable, quick peer-to-peer payments and rapid exchange transfers for arbitrage" [36].

Sharing several properties with Nano, a currently launched DAG-based cryptocurrency XDAG or Dagger [6] had each block contain exactly one transaction. At the same time, the block was an address. Among all transactions, the main chain with the maximum difficulty was allocated. In the main chain,

new coins were created about once a minute.

Compared to the $txDAG$ DL, there are also several block-based DLs, which are usually called $blockDAGs$. For example, Y. Sompolinsky et al. presented SPECTRE [45], PHANTOM and GHOSTDAG [46] by combining block with DAG data structures. The first one was designed for the consensus core of cryptocurrencies to achieve high throughput and fast confirmation times while remaining secure [29]. The other two focused on protocol scalability and leveraged PoW as consensus for a permissionless ledger [46]. All these blockDAGs generalized Nakamoto's blockchain to a direct acyclic graph of blocks.

PoW consensus was an innovative invention, while with its own drawback of intensive computation requirement no matter used in blockchain, $txDAG$ or $blockDAG$. Recently, to breakout the limitation, Z. Zhang et al. [57] proposed a new consensus mechanism named Proof of Authentication (PoA) based on the security and certificate properties of named data networking. Using the proposed consensus, the authors further introduced an IoT-Friendly private DL system, DLedger [58], which was based on DAG and motivated by IOTA. This NDN-DAG combination design provided us a good way to extend DAG applications and development of new consensus. However, since DLedger was very new and not examined by enough users, we decided to use IOTA as our system design protocol. After all, IOTA was usually seen as the most suitable decentralized protocol for IoT scenarios so far in terms of scalability, throughput, security and the feeless property, etc.

Chapter 3

Technical Overview of IOTA

IOTA [52] is the first DAG-based open-source DL which claims to "power the future of the Internet of Things with feeless microtransactions and data integrity for machines". In this section, we describe IOTA from several technical perspectives.

3.1 Network and Nodes

According to the access and permission control, networks can be divided into private and public networks. The public IOTA network is usually considered as the Mainnet responsible for processing all IOTA cryptocurrency transactions, with almost half of the nodes located in Germany. There are no access controls for participants to join public Mainnet so that anyone can run a node to read from and write into the public ledger. The private DL network generally requires a permission management mechanism such as the membership service provider (MSP) [48] in HyperLedger Fabric network. By contrast, IOTA foundation encourages people to continuously and stably participate by running always online nodes. For current IOTA release, a private network relies on an open-source Coordinator, called Compass [9], to protect it against attacks and to confirm transactions. As for the permission management, there is no specific mandatory solution for IOTA so that all existing approaches can become

the candidates.

Either in the public or private IOTA networks, there are two main types of nodes: Full Node and Light Node, see Figure 3.1. The Full Node (also called IRI node) maintains an entire ledger, and receives and validates transactions by running an IOTA Reference Implementation (IRI) instance in the background. Specifically, this IRI instance is in charge of connecting to neighboring nodes, tip selection, validating, broadcasting and synchronizing transactions with the Tangle. In contrast, the Light Node sends transactions to Full Node and requests services, acting as a client. Thereby, in order to participate in the IOTA Peer-to-Peer network, a Light Node needs to connect to a Full Node. As for the PoW, it can be performed either on the client-side or on a remote PoW service provider through an API. It should be noted that all Light, IRI and Client nodes mentioned here are just different roles in IOTA. In practice, they can be installed together in any physical or virtual machine with sufficient resources.

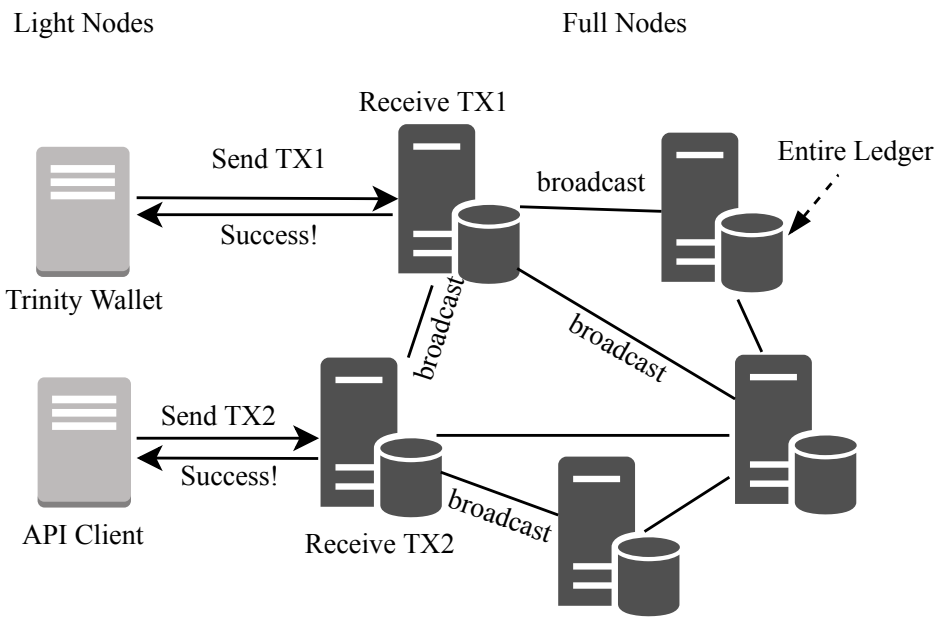

Figure 3.1: IOTA Peer-to-Peer network

3.2 Data Structure

First, it is worth noting that there is no concept of block in IOTA. Every single transaction is directly attached to the DAG data structure as a particle in this graph, see Figure 3.2. This block-less design can theoretically improve transaction efficiency by cutting the time of wrapping transactions into a block as in standard blockchain system and changing write policy from linear attachment in a chain to parallel way in a graph.

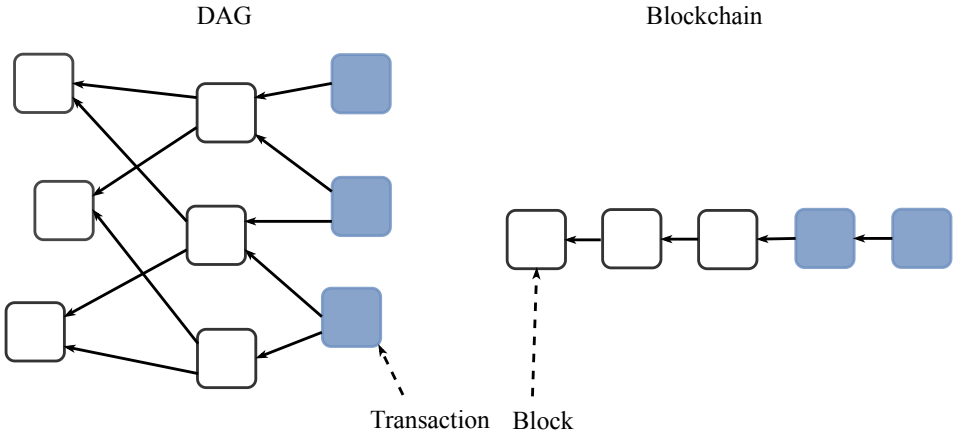

Figure 3.2: DAG vs Blockchain

Second, each transaction particle in the DAG records the complete historical information about this transaction in a list of data segments, as shown in Figure 3.3. Compared to standard Binary 1 byte giving $2^8 = 256$ possible data values, IOTA uses Trinary in which 1 tryte is 3 trits providing $3^3 = 27$ possible data values to store its transaction data. In IOTA, one transaction is composed of 15 data segments with 2673 trytes in total.

3.3 Transactions and Bundle

Transactions. In IOTA, a transaction is the fundamental operation that can stand alone. It can only send data without any value, or be packaged with other transactions. There are two basic types of transactions: Input transac-

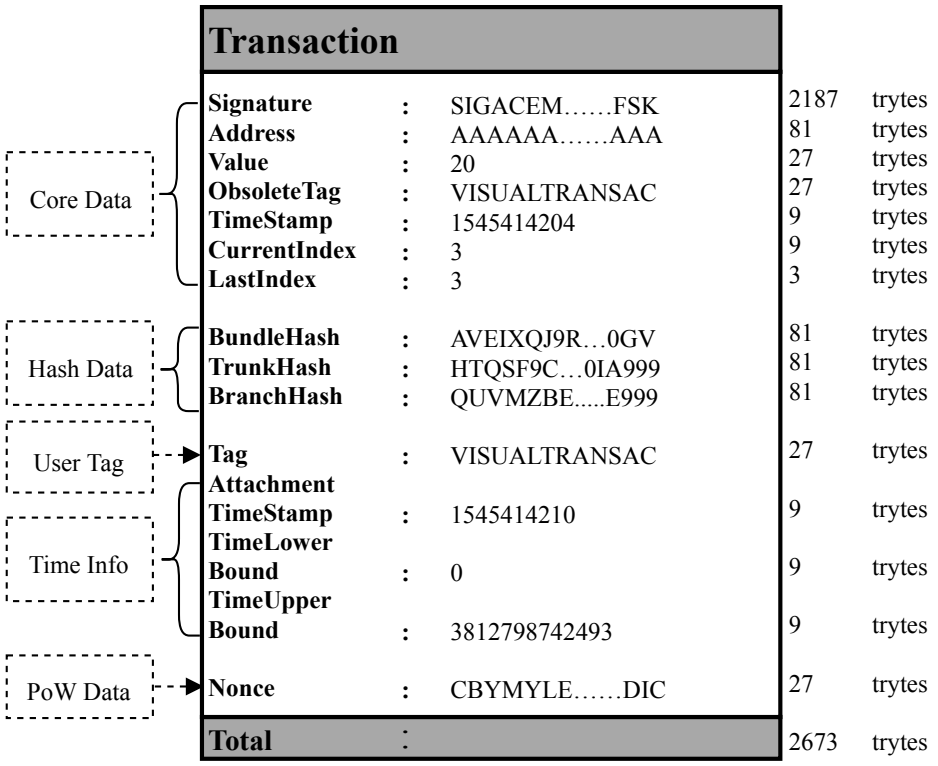

Figure 3.3: IOTA Transaction Data Structure

tion and Output transaction. Input transaction withdraws IOTA tokens from an address of the sender and contains the signature that signs transactions to prove their ownership [51]. In the case of high security level with a very large signature, it is fragmented over zero-value output transactions in the bundle. Output transaction is in charge of depositing IOTA tokens into a recipient's address [51]. If the input absolute value is bigger than output, there will be an extra change transaction with positive value to put the change back into one of the new addresses of the sender.

Bundle. A bundle is a group of one or multiple related transactions. It acts as a transactions container in IOTA to transfer data or tokens. It is always an atomic operation, i.e., either all transactions are successful or none. The structure of a bundle consists of Output, Input, and Change transactions, which

are called head, body, and tail, respectively. Typically, the tail is $Index0$, and the head is the last transaction in the bundle, see Figure 3.4. All transactions, starting from the tail, are connected by reference to the one with the next index. These connections allow nodes to reconstruct bundles and validate their contents.

In order to attach transactions to the ledger, the head transaction needs to be connected to "the tails of two other bundles in the Tangle, and the tail and body transactions are connected to one of those tails as well" [50]. Moreover, these tail transactions are selected by IRI nodes through tip selection.

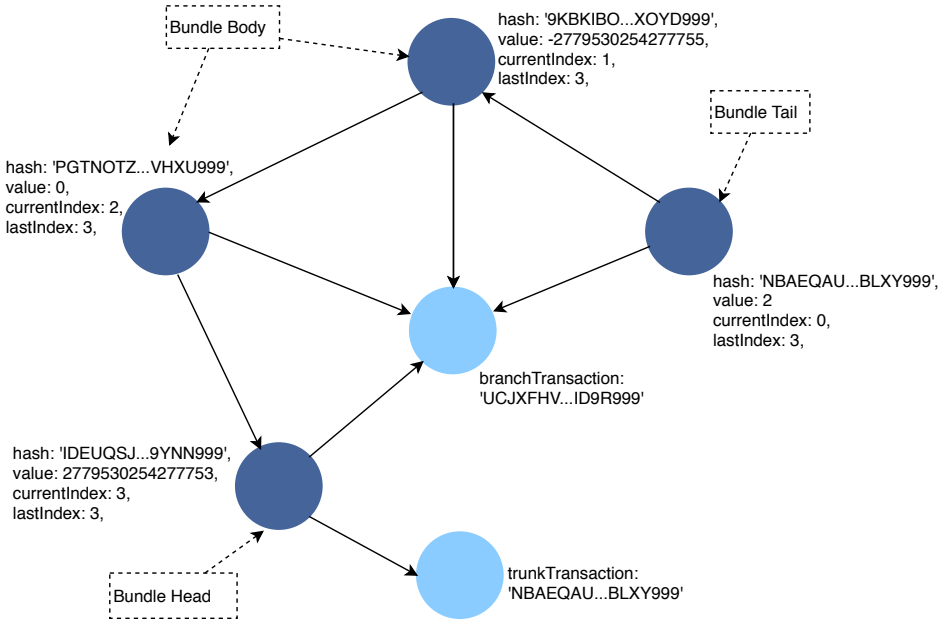

Figure 3.4: Bundle Structure

Transaction Flow. As previously mentioned, there are no miners in IOTA. As such the process of making a transaction is different from any Blockchain. The process flow to complete a transaction in IOTA is shown in Figure 3.5.

1. *Generate Bundle Hash:* During preparation for the Output and Input

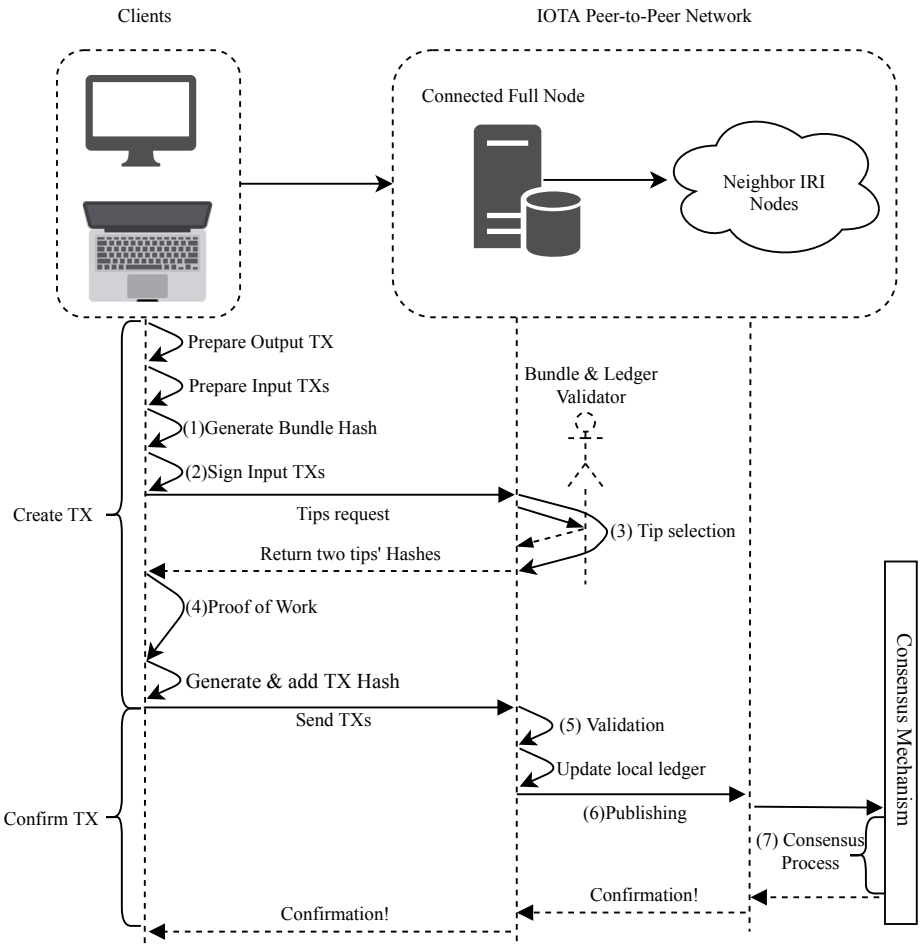

Figure 3.5: IOTA Transaction Flow

transactions, users can get all transaction validation items including address, value, obsolete tag, timestamp, index, and last index in a bundle. Then, the client uses Kerl hash function with sponge constructor to absorb transaction validate item one by one to generate the result of bundle hash.

2. *Sign Input Transactions:* All hashing in this step uses the Kerl hash function which converts between trinary and binary, and calls the well known Keccak hash function. There are two main sub-steps, generating the private key and signing the transaction [19]. First, a subseed is generated from the user's seed by taking the Kerl function to ensure it

is independent enough from the original seed. Then, the private key can be obtained from a key generator by successively hashing this subseed 27 times per security level. Second, one can use signature fragment generator with a private key and the above-mentioned bundle hash to get the transaction signature. Remember that there are in total 81 trytes in a bundle hash. Depending on the predefined security level value (S = 1, 2 or 3), the client will take the first $S \times 27$ (i.e., 27, 54, or 81) trytes, respectively, of the bundle hash, combined with the private key to generate a transaction signature for each transaction. Since the signature is hashed 27 times per security level, it results in a key length of $S \times 27 \times 81$ trytes, i.e., 2187, 4374, or 6561 trytes, respectively. Each 81-tryte hash generated constitutes a key fragment [19] in a transaction. After this step, all signatures are filled in the corresponding transactions.

3. *Tip Selection:* MCMC is used to randomly select two unvalidated transactions, called tips, from the local ledger in a Full Node. A client can get two tips from IRI node through the $getTransactionsToApprove$ API call. It is worth noting that transaction validations occur at both bundle level (e.g., if bundle values exceed global supply, inputs and outputs are balanced, and signatures are valid) and ledger level (e.g., if double-spend exists) during the tip selection. Finally, two tail transactions of previous bundles are referenced by the approver; and their hash values are filled into $branchTransaction$ and $trunkTransaction$ in the transaction's data segments. Section 3.5 discusses the MCMC algorithm in more details.

4. *Proof of Work:* In order to have the proposed transaction accepted by the network, some PoW similar to Hashcash, but not Bitcoin (spam and sybil-resistance) is required in IOTA. This can be done on the client-side, Full Node, or a third-party PoW service provider via a nonce search

algorithm called pearlDiver. According to the pre-set PoW difficulty level (Minimum Weight Magnitude, MWM), it usually takes from a few seconds to minutes on a standard PC. Once the PoW is completed, a condition-satisfied Nonce is generated and filled into the data segment. All transaction's generation steps are finished until the transaction hashes are generated and filled into transactions.

5. *Validation:* Between receiving and publishing a transaction, a Full Node needs to validate two preceding transactions previously selected via MCMC. The validation items are checked by the transaction validator, e.g., if the PoW is completed, if bundle value exceeds the total global supply and if the last address trit is 0 for value transactions. If every item passes the examination, the Full Node will update this new transaction to the local ledger by attaching it to the Tangle, and then synchronize it to its neighbors.

6. *Publishing:* The Full Node publishes the validated transaction to the whole network through neighbors and waits for it to be confirmed. IRI nodes communicate with their neighbors through a gossip protocol.

7. *Consensus Process:* After the transaction is published to the Tangle, its cumulative weight will increase as time goes by according to the MCMC. Thus, more and more new transactions will tend to validate it directly or indirectly until it gets confirmed. Then, an updated status will feed back to the client, and the whole transaction is complete.

3.4 Consensus Process

Any DL system needs a consensus as its core playing rule to define what a confirmed transaction is. In IOTA, this consensus has two versions: the cur-

rently running COO and the proposed confirmation confidence based one.

COO Confirmation Consensus. This is the current running consensus in the IOTA mainnet. Coordinator is a powerful node controlled by the IOTA Foundation, which is always regularly issuing a special kind of transactions (called Milestones) with zero values to validate transactions and secure the network. It is simply defined that any transactions with a reference from a Milestone are considered confirmed; and others are not.

When a Milestone comes to the Tangle, it traverses through ancestors (all transactions on its validation path) until it reaches a recent memory-persisted snapshot of balances, creates a new potential state, and then checks to see whether any addresses resolve to a negative value. Therefore, there is enough time for the new transactions to detect the conflicting transactions before the next Milestone under the MCMC mechanism.

COO-less Consensus. COO-less consensus involves confirmation confidence, which leverages fuzzy logic to convey a confirmation degree rather than binary confirmed or not. For a specific transaction X, the calculation of its confirmation confidence is determined using the following steps:

1. Use the MCMC method to select 100 new transactions (*tips*).
2. Calculate how many tips will directly or indirectly reach the transaction.
    - if it is less than 50%, the transaction is not yet validated (*not confirmed*).
    - if it is more than 50%, the transaction has a fair chance to be validated (*partially confirmed*).

Table 3.1: IOTA Consensus Comparisons

Metrics	COO	COO-less
Fuzzy Logic	No	Yes
Double-spend Protection	Yes	Yes
Efficiency	Medium (COO is Bottleneck)	Low
Domination Attack Protection	Yes	Yes (Under Big Network)

- if it is 99% or 100%, the transaction is considered validated (*fully confirmed*).

This consensus relies on a threshold to define confirmation. When the confirmation threshold is set to be extremely low, the consensus is considered to be simplified. Moreover, it will take much time and energy to run MCMC for many times in order to calculate the confidence. If each transaction updates its confirmation confidence for every new tip's attachment, this secure consensus will lead to a low confirmation efficiency.

By analyzing and comparing different consensuses (see Table 3.1) in IOTA, there appears to exist a trade-off between confirmation speed and double-spending hazard in IOTA.

3.5 MCMC Random Walk Tip Selection

In IOTA Tangle, the number of tips increases [3] as new transactions keep adding to the ledger. It becomes crucial how to wisely select two tips from all the valid tips so that the whole Tangle can grow in a "healthy" way. Before developing a solution, some basic notation needs to be reviewed.

EntryPoint. There must be a starting position for any walk to start the

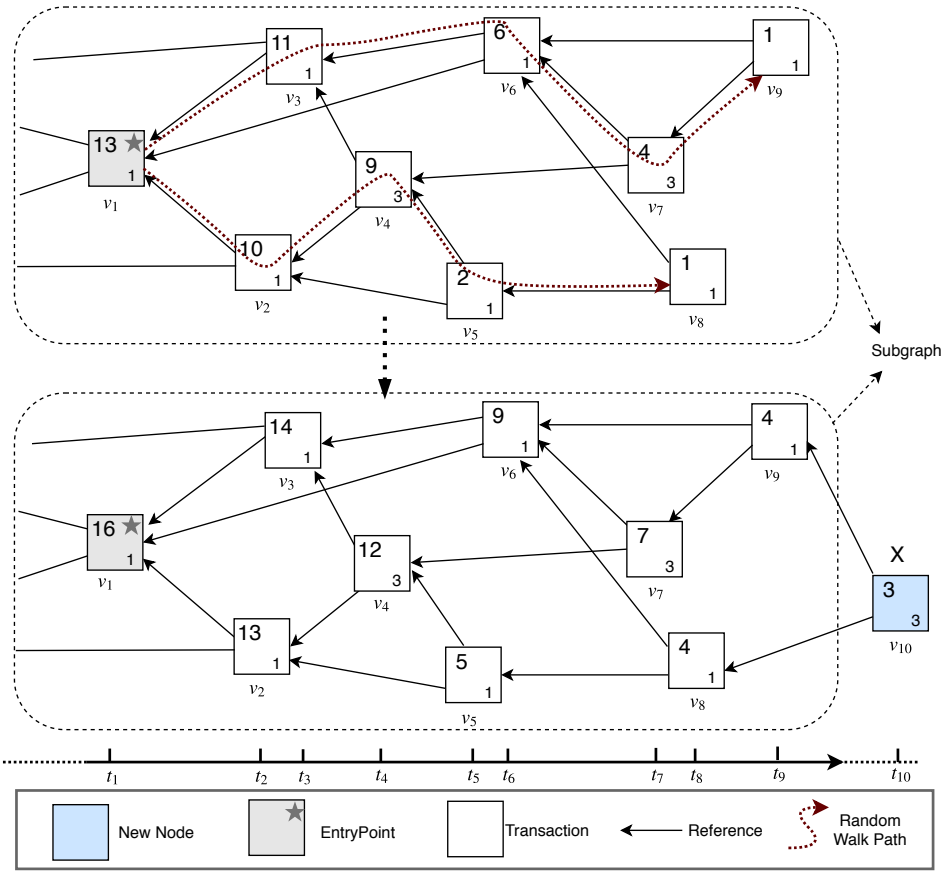

Figure 3.6: MCMC weighted random walk

random walking. It is easy to think of taking the genesis node as the entry point. However, this will bring obvious efficiency decrease while traversing a long path when the graph grows large. In the COO version IOTA Tangle, a walker takes the latest solid Milestone and recurs a pre-specified n depth (e.g., n equals 2 in Figure 3.6) number of Milestones in the past. The DAG starting from this old Milestone is called subgraph. Moreover, this old Milestone located n depth ago is the entry point. Based on the experimental results from [25], any entry point placed further than 10λ to 20λ (transaction arrival rate) provide the same growth of the number of tips. So, one can empirically conclude that such starting position does not directly influence the tip selection results. However, we need to balance the security and efficiency for tip selection.

Cumulative Weight. For each transaction in the subgraph, there is a property called cumulative weight, defined as the summation of its weight and the sum of weights of all transactions that directly or indirectly refer to this transaction. The weight of a transaction reflects the computation resource that the sending node puts into this transaction. As a new transaction comes to the tangle, all previous transactions connected to it in the subgraph updates by adding the new weight. For example, in the lower part of Figure 3.6, every transaction connected to X increases its weight by 3 after X is attached. However, updating the cumulative weight too frequently will decrease the efficiency.

MCMC Weighted Random Walk. Having the entry point and current Tangle state of cumulative weights for all transactions, the walker walks through the subgraph twice, for each time from the entryPoint to reach a tip, so as to get two selected tips. For example, the selected tips in Figure 3.6 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. Specifically, the walk chooses next approver transaction in a probability determined by the cumulative weights and a randomness parameter, called α.

The tip selection algorithm never ends inside a bundle even in the case that the rest of the bundle has not yet arrived at that node. If the selected tip is not a tail in a bundle, the MCMC will "walk back" to the previously visited tail transaction.

Chapter 4

DAG-based DL Application in Smart Communities

According to the Smart Communities Guidebook, developed by the State University of San Diego (1997) [22], smart community is a geographical area in which residents, organizations, and governing institutions are leveraging information communication technology (ICT) to change their region in significant ways. The concept of smart communities is used all over the world with different nomenclatures, contexts and meanings. However, from the technical perspective, as R. Lea [27] pointed out, it is critical for smart communities to have underlying communication platform which enables smart communities to connect infrastructure, devices, and people. In other words, a smart community must rely on an efficient and secure communication platform to share information and resources, and transfer values with each other. For example, as the renewable energy, smart and micro grid are rapidly developing, there is an increasing need for local community residences to trade their distributed energy resources. Traditional centralized solutions usually get third-party involved and bring many concerns such as privacy, security, scalability and high system maintenance fees. Therefore, a decentralized, scalable and secure solution is needed for a smart community to transfer values and resources, e.g. distributed energy resources transactions.

4.1 DAG-based Smart Homes Design

In a NoSQL database system, the CAP-Theorem postulates that only two of the three different aspects, i.e., strong consistency, availability and partition-tolerance, can be fully achieved at the same time [24]. Likewise, there is a well-known Blockchin Trilemma in designing distributed ledger systems. According to Buterin [20], the founder of Ethereum, a BC platform can only fundamentally achieve 2 out of the following 3 traits at one time:

- **Decentralization:** A system is running with each participant node only having access to limited computation, bandwidth and storage resources, i.e., no single node or group has access to majority of resources to manipulate the whole network.
- **Scalability:** The system's power of processing transactions must increase to handle a scenario of mass of users as the network scale increases.
- **Security:** Can handle all attacks from any entity with less computational resources than the system itself.

Currently, blockchains such as Bitcoin and Ethereum are designed to focus on decentralization and security, with the expense of scalability. The underlying reason is that all full nodes in these respective peer-to-peer networks must reach consensus before transactions can be added to the ledger. As the network scale increases, more time is needed to reach consensus, i.e., the majority of network nodes validate the transaction and agree on adding it to the ledger, which leads to slower transactions speed.

However, we argue that this trilemma can be addressed in our case by leveraging the architectural design principle of separation of concerns (SOC).

Specifically, we achieve decentralization and scalability by using the IOTA consensus, and utilize a decentralized coordinators design with permission management to meet the security requirement. We describe the proposed solution in terms of architecture, consensus mechanism, coordinators and permission management.

4.1.1 Architecture

Figure 4.1 illustrates a fundamental high-level picture of the proposed architecture. There are three main parts including smart homes, the Tangle of inter-house transactions (TXs), and smart devices in the homes. In each smart home, there is an always online computation device called home node with preinstalled firmware and corresponding tangle reference implementation. Every home node is connected to its neighbor nodes with TCP/UDP protocols for communication and synchronizing the distributed ledger. In practice, this home node can be any kind of IoT device or specialized chip, that can provide the computational power, such as a server, VPS, PC or even microcomputers like RaspberryPi. All home nodes in a community provide the computational power to maintain and secure the distributed ledger network.

When this system is used in the case of energy transaction, we assume that there is a microgrid as the infrastructure behind the home nodes network. The microgrid connects all photovoltaic systems and other distributed energy resources (DERs) installed at smart homes. Our proposed solution provides the microgid with a payment system to carry out DERs inter-house transactions in a decentralized, efficient and secure way without any transaction fees in a local smart community. In fact, the Tangle with DAG data structure can act as a data management system to transfer, store and even query both inter-house and intra-house data. For example, it can be used to send, receive

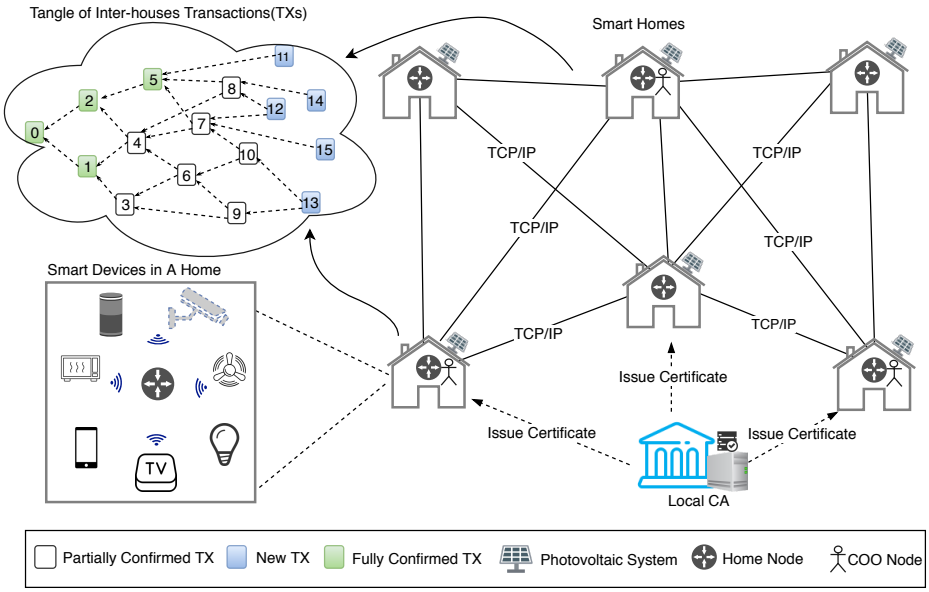

Figure 4.1: DAG-based Smart Homes Architecture

and store the command message to remotely control smart devices in a smart home. However, in this paper, we just focus on the scenario of inter-house transaction data including sending, receiving and confirmation transactions, e.g. energy transactions.

4.1.2 Consensus Mechanism

In a payment system, there is always a mechanism called consensus which identifies when a transaction can be securely considered confirmed and added to the ledger. For the consensus mechanism of our solution, we follow the IOTA reference implementation and employ IOTA tangle to handle all transactions. In this system, a new transaction must select two previous unapproved transactions, or tips to approve according to a tip selection algorithm before adding to the Tangle. Basically, there are two solutions to reach consensus in the tangle, as seen in Section 3.4. We choose the currently implemented coordinator (COO) [23] as the consensus in our system design.

In Figure 4.1, green boxes are fully confirmed transactions, which indicates that they are validated by all of current tips, while the white boxes are only partially confirmed. The blue boxes represent tips without any validation. In current public IOTA project, the coordinator is an entity controlled by the IOTA Foundation, which generates a zero-valued transaction, called a milestone, every minute. Under the coordinator consensus, only transaction referenced by a milestone can become confirmed, while the others cannot.

4.1.3 Coordinators

As the IOTA Foundation explains, the Tangle network has a small number of nodes at its infancy stage so that an attacker can easily create a lot of nodes and thus creating many malicious transactions. According to the MCMC algorithm, there is a relatively large chance that these malicious transactions are selected to be confirmed. To protect the Tangle in its infancy from 34% attack, a protection mechanism called the coordinator (COO) is employed. This does not mean it is centralized because the COO node follows all the consensus rules just like any other node. The only activity performed by COO node is continuous generation of trustable transactions which contain zero values, to help secure the infancy Tangle network.

In our solution, we modify the COO mechanism to be a cluster of randomly chosen COO nodes which are the normal home nodes equipped with the ability of issuing milestones, as shown in Figure 4.1. Therefore, if any COO node crashes, others can continue to take the responsibility and create milestones to confirm the transactions. This decentralized design not only removes the single point of failure from the system, but also reduce the risk of centralization.

4.1.4 Permission Management

To build the trust for all network participants, a hybrid security solution combining the permission management system and proof of work is employed before the number of nodes is large enough. Motivated by Membership Service Providers (MSP) solution in Hyperledger [21], we propose a similar local certificate authority (CA) using X.509 certificates as the permission management system. In a local community, each home node needs to get certification issued by the local CA to join the network, as shown in Figure 4.1. It is worth noting that the CA is in charge of initial certification and new node's access control. When it fails because of some unexpected issues, the existing DL system can still work under an untrusted environment.

To summarize, we propose a payment solution for smart homes peer-to-peer local energy transactions. Specifically, we leverage the DAG-based distributed ledger technology to build a permissioned, private and secure transaction network. In the next section, we evaluate and analyze some important metrics such as transaction speed and scalability of our proposed solution.

4.2 Evaluation and Analysis

In this section, we will describe our experiments and results, and evaluate our proposed solution in terms of the transaction speed and scalability. Then, we conduct a general analysis and discussion based on the results, providing some important insights about the DAG-based DL network application in IoT.

4.2.1 Experiments and Results

We employ IOTA Implementation Reference (IRI 1.5.3) and a coordinator simulation tool to deploy a private IOTA network on the SAVI OpenStack

cloud platform [1]. In total, 40 nodes with the flavor of medium size virtual machines (4GB RAM, 2 VCPU and 40.0GB Disk) are used to build a network. We choose the medium size rather than high performance nodes because this is more likely to fit the IoT scenarios with a low hashpower. In fact, more powerful nodes will improve the performance by reducing the time of proof of work and thus increasing the transaction speed. In practice, these nodes represent the home nodes installed in the smart homes.

In order to explore the metrics of transaction speed and scalability, we design

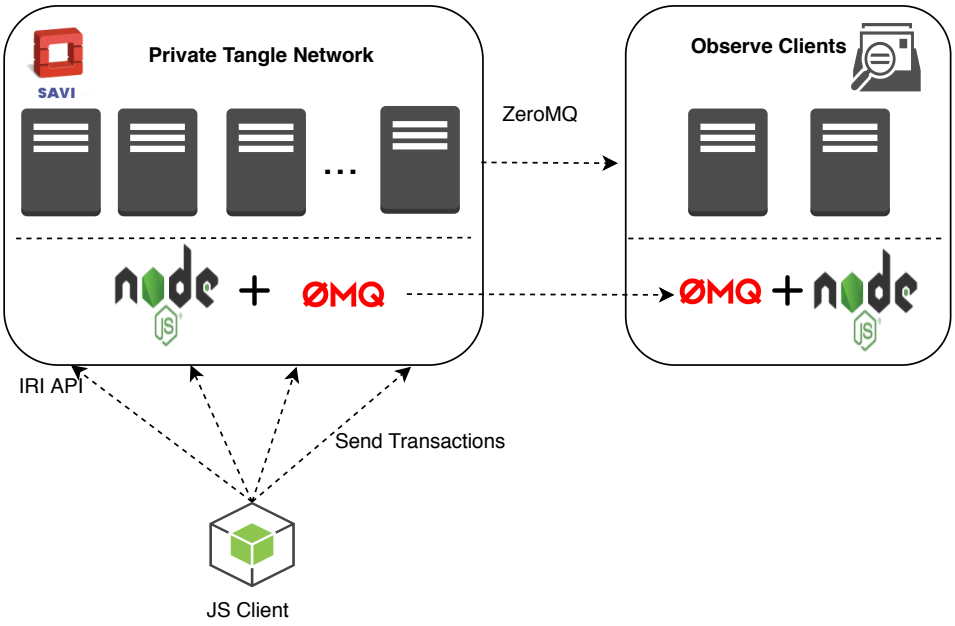

Figure 4.2: DAG-based Smart Homes Experimental Setup

a load testing method that consists of sending and receiving parts. For sending component, we set each sending node or sender to intensively send transactions in every short time interval such as 1 second or 2 seconds depending on the difficulty of proof of work called Minimum Weight Magnitude (MWM). All these transactions will be broadcast to all nodes through TCP/UDP. We control the transaction sending rate by controlling the number of senders during a test time window. For receiving component, we leverage zero message queue

[1] https://www.savinetwork.ca/

(ZMQ) to listen to the specified port on a home node and receive transaction data by subscribing *tx* and *sn*, which indicate all received transactions and new confirmed transactions, respectively.

We test the transaction speed of both TPS and CTPS under different network node scales (10, 20, 30, 40) with different *MWM* configurations, as shown in Figure 4.3, 4.4, 4.5 and 4.6, where TPS refers to the number of received transactions per second and CTPS refers to the number of confirmed transactions per second in the tangle. Three coordinators are randomly selected and set to generate milestones every minute.

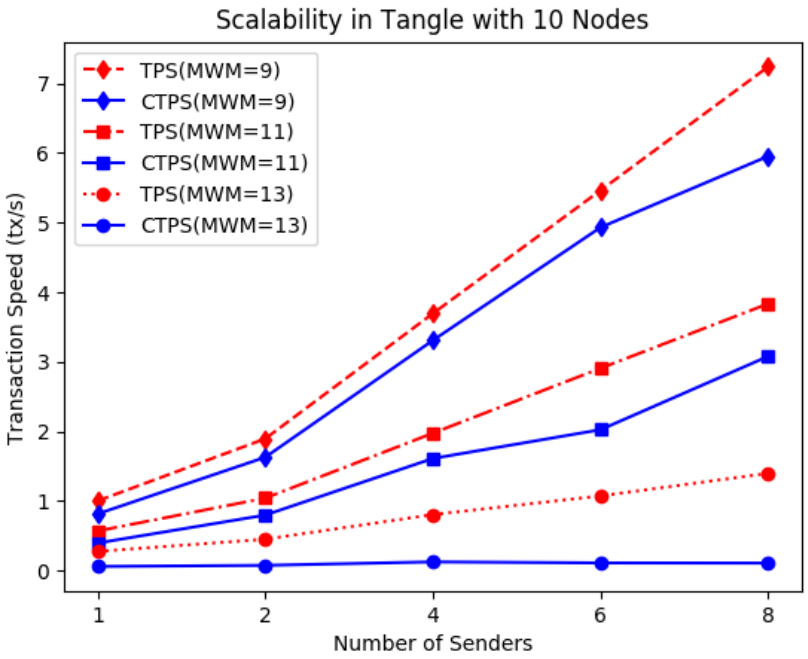

Figure 4.3: Scalability in 10 Nodes Network

In order to explore if different network sizes influence the transaction speed, we use the same 10 senders with 3 COOs to test the TPS/CTPS under 10, 20, 30 and 40 nodes networks, respectively. The result is shown in Figure 4.7.

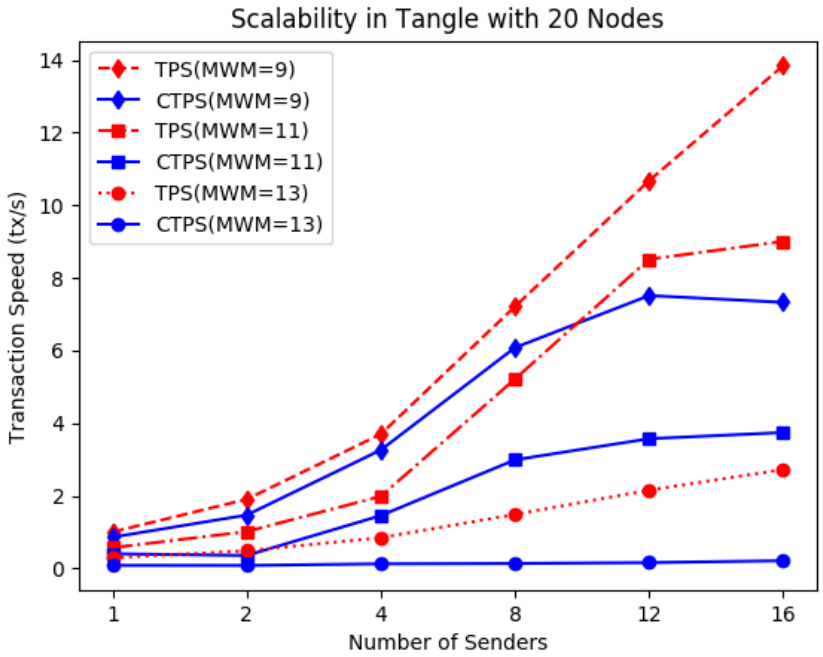

Figure 4.4: Scalability in 20 Nodes Network

In current implementation, COOs play a critical role for confirming transactions. Therefore, we conduct an experiment to test the effect of different numbers of COOs on transaction speed by changing the number of senders under 40 nodes network and keeping MWM=9. The result is shown in Figure 4.8.

4.2.2 Analysis and Discussion

In this part, we discuss our proposed solution from the system throughput, scalability, decentralization and security perspectives by analyzing the experimental results.

Throughput: from the load test results, our solution provides a relatively good efficiency of processing transactions, as shown in Table 4.1. Even

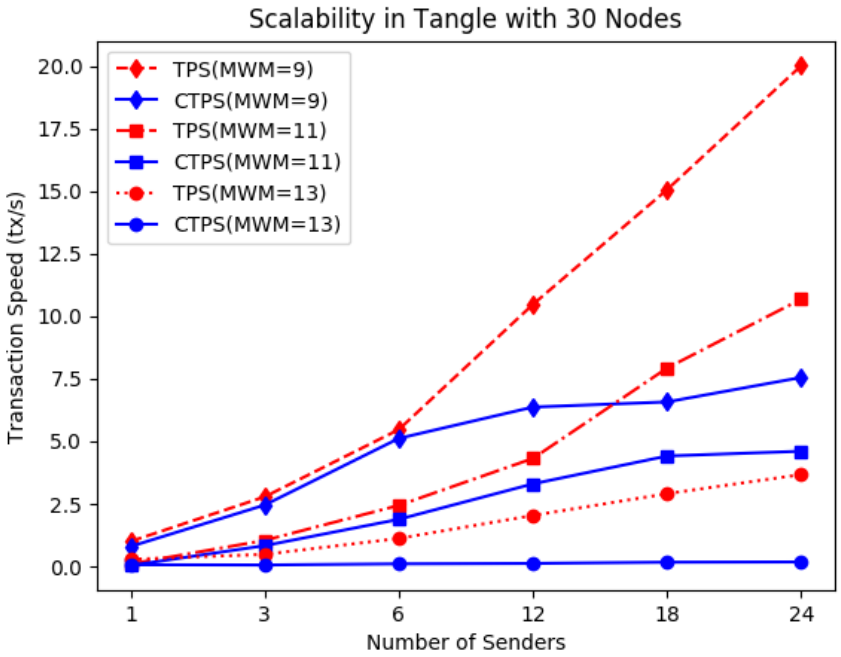

Figure 4.5: Scalability in 30 Nodes Network

Table 4.1: Basic Performance Statistics in 40 Nodes Network

Performance Values Under Various MWMs			
Items	MWM=9	MWM=11	MWM=13
Send Interval(s)	1	2	4
Max TPS(tx/s)	26.26	14.69	4.84
Max CTPS(tx/s)	7.13	8.34	0.19
Min TPS(tx/s)	1.00	0.57	0.30
Min CTPS(tx/s)	0.83	0.38	0.05

in the situation of only one sender, the average TPS and CTPS can reach 1 tx/s and 0.83 tx/s, respectively, as shown in Figure 4.6. This is good enough for the scenario of inter-house energy transaction in a local community.

From Figure 4.7, many flat lines tell that the nodes scale of network has almost no influence on the transaction speed. From Figure 4.8, we find that the CTPS has a peak value at a proper level of COOs number, e.g. CTPS reaches around 7.5 tx/s under 24 senders and 6 COOs. According to the consensus of currently implemented IOTA version, a transaction must refer to two tips

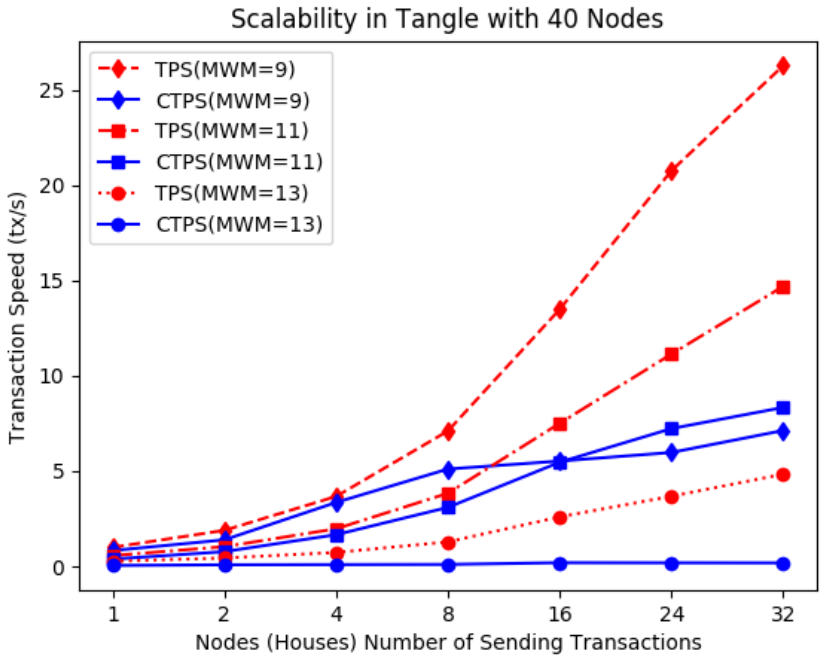

Figure 4.6: Scalability in 40 Nodes Network

and it can be considered confirmed when any milestone directly or indirectly reaches it. If not enough milestones are created, some transactions may not get any references to be confirmed. If too many milestones are created, it may lead to a lower transaction generation rate in a closed network because of the hashpower competition. Therefore, there should be a balance between the number of COOs and the number of unconfirmed transactions in the Tangle at one time.

Scalability: from the TPS/CTPS results under different networks (Figure 4.3, 4.4, 4.5 and 4.6), it is obvious that as the number of senders increases, the transaction speed of both TPS and CTPS almost increases linearly. This indicates that transaction speed has a good linear scalability against the number of senders.

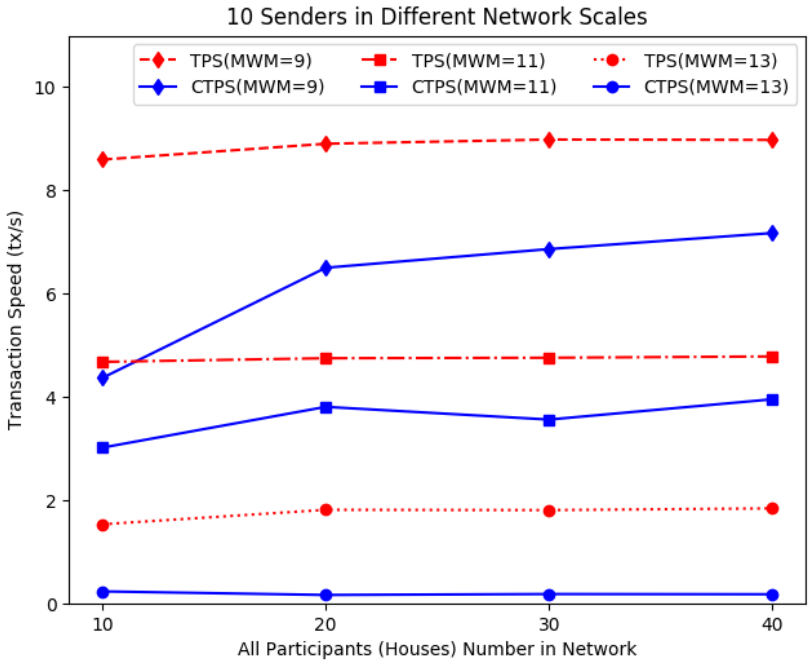

Figure 4.7: Scalability in Different Configurations

Decentralization: in our proposed solution, each home node only has access to limited computational resources. The decentralized coordinators are randomly selected from the network participants and independently confirm transactions. There are no central nodes and middle man roles in transaction processing.

Security: IOTA consensus provides the protection for double spending attacks. From system design perspective, on one hand, the permission management system sets a trustable and private environment for transactions like the first wall. On the other hand, all the home nodes with changeable PoW difficulties will build a hashpower wall (the second wall), which combines with the decentralized COOs to protect the system from 34% attack.

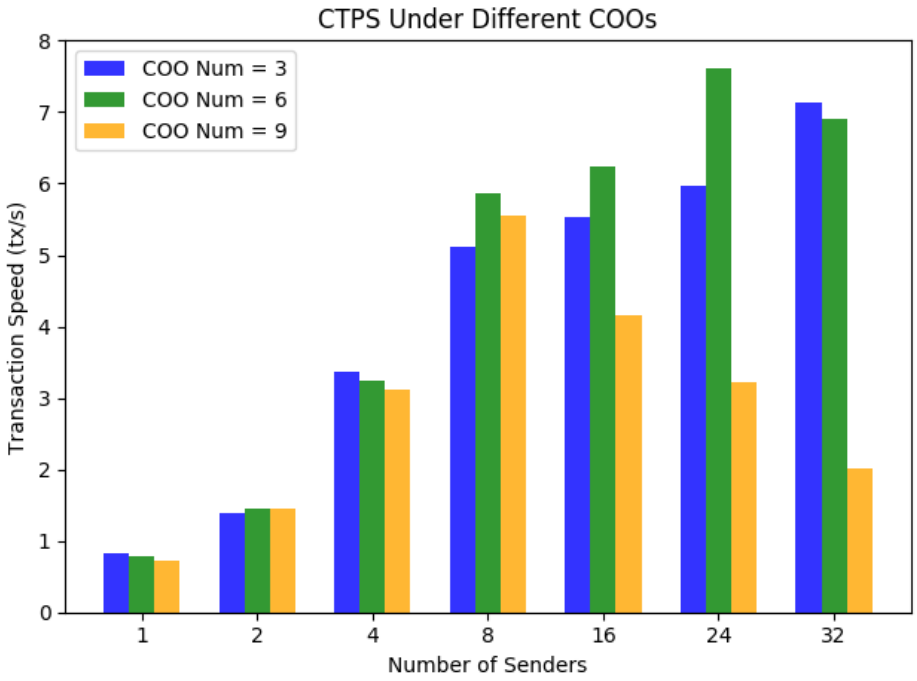

Figure 4.8: Transaction Speed Under Different COOs

4.3 Summary

In this part, we have proposed a DAG-based distributed ledger solution for IoT. Specifically, we introduced our solution in the context of smart homes for handling inter-house DERs transactions. With the initial experimental results, our solution provides high TPS/CTPS and good scalability which is suitable for IoT applications. We conclude that our proposed DAG-based distributed ledger is an effective solution for building a smart home IoT infrastructure. Further studies will be conducted on the decentralization optimization, security verification and containerization.

From all the analyses of experimental results, it is obvious that the proposed private DL system has good scalability and high transaction processing performance. By analyzing the principles of IOTA protocol and the insights of experiments, we can summarize that the scalability benefits from the efficient

MCMC consensus algorithm and the parallel DAG data structure for attaching and storing transactions, as well as the well-designed ledger synchronization mechanism.

However, the system performance still remains to be specific and empirical, which relies on the deployed system or at least a prototype system. We cannot answer the performance question such as on the throughput (confirmation rate), the average waiting time and the transaction confirmation ratio. Therefore, we will conduct a comprehensive performance analysis for this DAG-based DL system in next chapter.

Chapter 5

Performance Modeling

In this chapter, we will propose a hybrid performance modeling to evaluate the IOTA system performance. We strive to answer the two following vital questions on IOTA performance:

1. Which factors influence the throughput of an IOTA system? And how, quantitatively, do they impact the throughput?

2. What would be the optimal waiting time to wait for confirmation before reattaching the same original transaction?

All research described in this contribution has been conducted within the context of private IOTA network designed for smart communities. Two models, a simulation model and an analytical layered model, are proposed to answer two different performance questions, i.e., system Throughput and RWT, respectively. To conduct the simulation, we leverage and extend DAGSim simulator [56]. The analytical layered model is proposed to explore the confirmation distributions in the Tangle [40]. In addition to the analysis of simulation results, all results are experimentally validated.

5.1 Performance Metrics

5.1.1 Throughput

Throughput is a system level performance metric which defines how many jobs can be processed per unit time. In our proposed DL system, the throughput is defined as the confirmed transactions per second (CTPS), which represents the transactions processing power of the system. As for the definition of confirmation, it depends on the consensus used in the system, see Section 3.4 for details. Particularly, we choose COO as the consensus throughout this study, because this is the currently running consensus in IOTA.

5.1.2 Reattachment Waiting Time

Similar to a BC 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 Full 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 remain unconfirmed for a long time and be abandoned eventually. 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. So, too short RWT will not only waste power, but also cause network congestion due to amount of redundant transactions. On the other hand, too long RWT will dramatically increase confirmation latency and decrease user satisfaction.

5.2 Formalizing the Problem

In this section, we leverage graph theory to formalize the confirmation process in IOTA DAG so that we can solve it in a mathematical fashion. Basically, this is an ever-expanding Directed Acyclic Graph (DAG) as time goes on. In this graph (G), there are two types of nodes (V), which are Transaction Nodes (V_{T}) and Milestone Nodes (V_{M}). In practice, all new nodes are generated with the importance ratings called weights (h), and attached to the DAG with different arrival rates:

- For V_{T} nodes: according to a Poisson process with rate λ_{T}, also interchangeably used as λ;
- For V_{M} nodes: according to a constant arrival rate λ_{M}.

This DAG is expanding in a way that the later arriving nodes need to approve previous nodes. Here, approval is a relationship of direct reference. For example, in Figure 5.1, node A approves B, which means that A is directly pointing/referring to B.

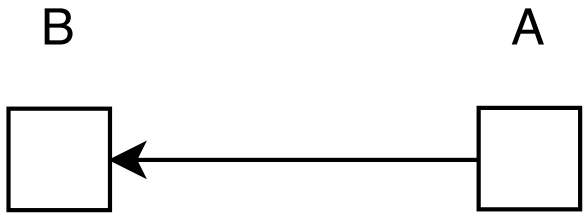

Figure 5.1: Approval Example

All the references in the DAG compose of the directed edges collection E. Therefore, in a specific moment, this DAG can be defined as,

$$G =< V, E > \tag{5.1}$$

where $V = V_{\mathrm{T}} \cup V_{\mathrm{M}} = \{v_1, v_2, \ldots, v_n\}$ denotes all nodes with the index based on time serials.

In order to join the DAG, any new issued node (no matter V_{T} or V_{M}) needs to approve two previous unapproved nodes, called tips. These two tips are chosen by following the tip selection algorithm, which is a random selection mechanism among the section close to tips. Specifically, this algorithm includes 3 steps:

1. **Define a Subgraph and EntryPoint:** take the latest solid milestone and recur a prespecified depth (e.g., equals 2 in Figure 5.2) number of milestones in the past. The DAG starting from this old milestone is called subgraph. This old milestone is called entryPoint (V_{M1} in Figure 5.2) for this walk.

2. **Rating Calculation:** for each node in the subgraph, calculate the corresponding cumulative weight, which is defined as the summation of the own weight of a particular node plus the sum of weights of all nodes that directly or indirectly approve this node, as shown in Figure 5.2.

3. **Random Walk:** walk through the subgraph twice, for each time from the entryPoint to reach a tip, so as to get two selected tips. For example, the selected tips in Figure 5.2 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.

Here, the small number in the lower-right corner of each box denotes own weight, and the bold number in the upper-left corner denotes the cumulative weight. According to the implementations of Rating Calculation (cumulative weight) and Random Walk (WalkerAlpha), the probability to walk towards a

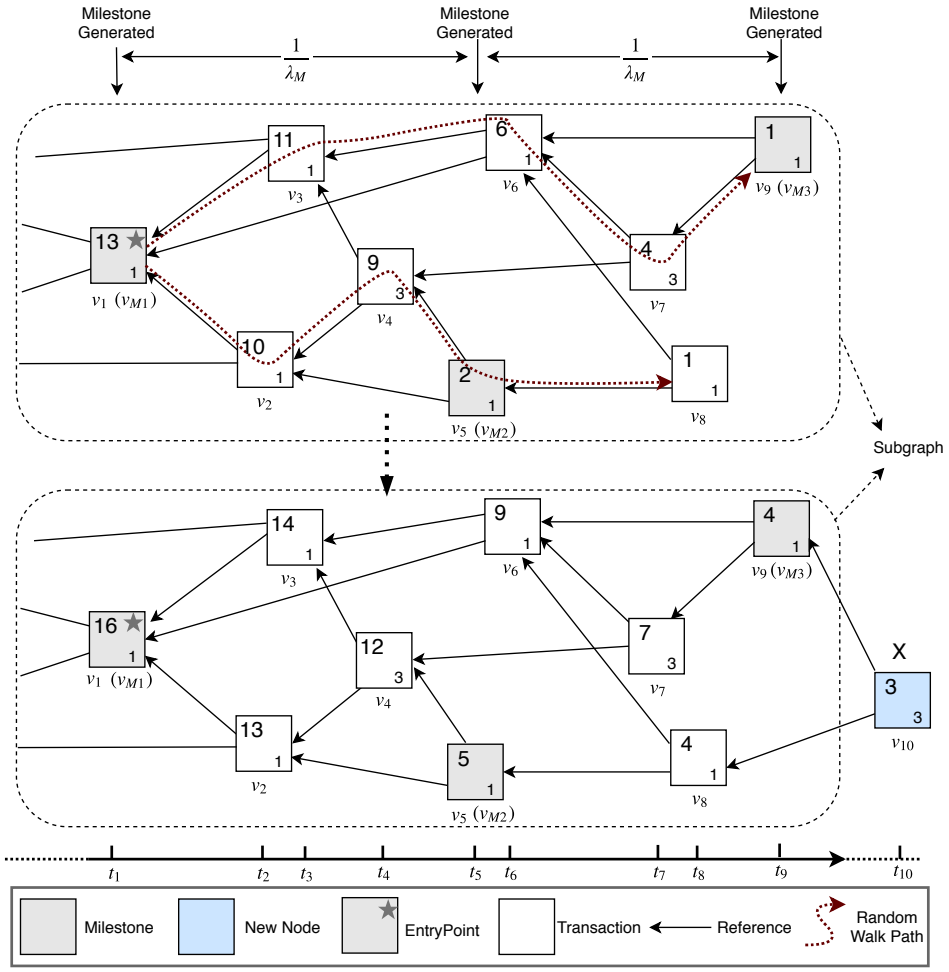

Figure 5.2: Subgraph with Weight Assignments before and after a Newly Issued Node

specific approver is as the following formula,

$$P_{\mathrm{xy}} = \frac{e^{\alpha H_{\mathrm{y}}}}{\sum_{z:z \rightsquigarrow x} e^{\alpha H_{\mathrm{z}}}} \tag{5.2}$$

where P_{xy} is the probability to walk from node x to y, H_{y} is the cumulative weight of node y, and $z \rightsquigarrow x$ means "z directly approves x". Therefore, in the random walk step, a node 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. Here, α is a randomness factor, which means that an $\alpha = 0$

factor makes the walk purely random; and with an $\alpha = \infty$, the highest-rated node will be always chosen as the next step. The sum in the denominator is a normalization factor, which sets the total transition probabilities sum to one.

Any node getting the reference (approval) directly or indirectly from any V_{M} (i.e., milestone node) is defined as confirmed, namely there is a path from a milestone to any confirmed node, while others are unconfirmed.

Question: Given the node arrival rate λ_{T}, λ_{M}, randomness factor of α and all the rules for tip selection, what is the average confirmation rate (confirmed transactions per time unit) in equilibrium?

5.3 Empirical Simulation Modeling

Simulation modeling is the modeling process that leverages simulator to generate and confirm transactions, then collect data to mine the relationship between performance metrics and configured parameters. A good simulator will help a lot to mock the real system's behaviors and reveal insightful characters about the system.

5.3.1 Simulation Model

COO consensus, based on the DAGSim simulator [56], has been developed to perform the simulations. Since DAGSim only supports COO-less consensus, no milestones are generated in between normal transactions. We extended this simulator to support both COO-based and COO-less consensus; we configured the extended DAGSim to generate and broadcast milestones to the network every 60 seconds. The generated milestones are acting exactly like normal transactions, but with the capability to confirm transactions. When we want to check whether a transaction is confirmed or not, we just simply check if

it is directly or indirectly referenced by a milestone. Our extended version of DAGSim is available publicly[1].

In 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)* shown in Algorithm 1. 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 in the case of 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 Genesis 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 confirmation rate can be calculated.

To collect the simulation data, we run a group of 10 simulations with $transactions = 6000, agents = 20, d = 1, \alpha = 0.001$ and λ varying from 1 to 10 with $step = 1$. This is a base-line configuration which we use to make comparison with others to explore the influence. Then, we run 5 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 rates 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; $transactions$ is set to 6,000, so that at least 10 $Milestones$ are ensured for each simulation for lower

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

Algorithm 1 Indirect References Extraction Algorithm

```
1: indirect_references = empty
2: function FIND_REFERENCES(tx, direct_references)
3:     APVD_1 = The 1st transaction approved by tx
4:     APVD_2 = The 2nd transaction approved by tx
5:     if tx is genesis then
6:         End
7:     else if tx is in indirect_references then
8:         Append the new TXs to indirect_references
9:         End
10:    else
11:        if APVD_1 is not in indirect_references then
12:            Append APVD_1 to indirect_references
13:        find_references(APVD_1, direct_references)
14:        if APVD_2 exists then
15:            if APVD_2 is not in indirect_references then
16:                Append APVD_2 to indirect_references
17:            find_references(APVD_2, direct_references)
```

λ values. The simulations are conducted on a DELL PC with Windows 10 OS, 8th Generation Intel Core^{TM} i7-8700 12-Core Processor and 16GB RAM.

After simulations, the proposed $IREA$ is used to extract the transaction confirmations data and conduct a statistical analysis on the data. The result provides an almost linear relationship, as shown in Figure 5.3, in which all CTPS values are the results obtained by averaging over all milestones confirmations.

In order to see if different tip selection strategies have an impact on CTPS, two more groups (10 simulations in each group) of simulations, non-weighted and uniform random tip selection (URTS), are conducted, respectively. In weighted random walk, to explore if the randomness factor α will influence CTPS, we conduct 3 other groups of simulations with α equals to 0.01, 0.1 and 1, respectively. Also, different network delays indicated as distances are examined. For each group simulations, the value of λ is varying from 1 to 10

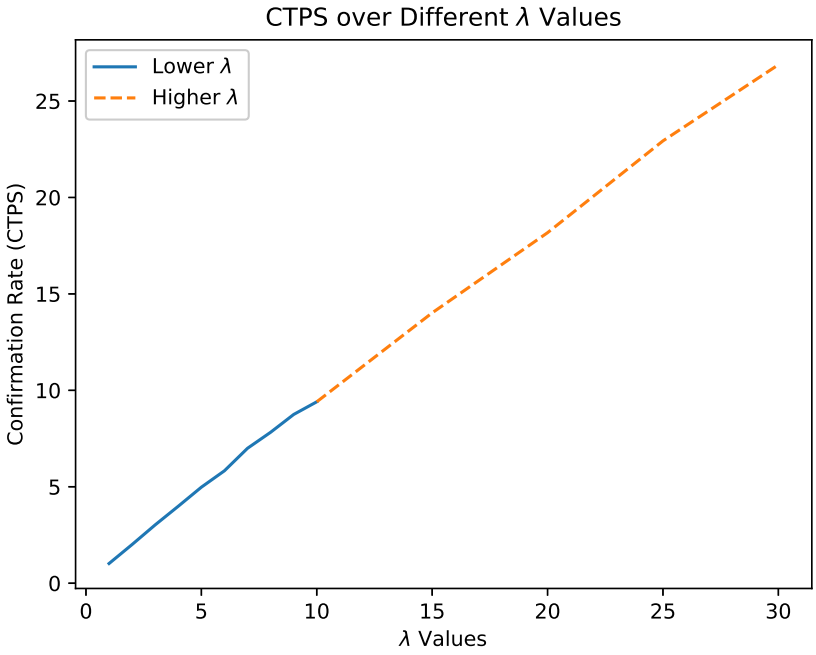

Figure 5.3: CTPS over Different λs in the Simulation Model

with $step = 1$, see Figure 5.4, Figure 5.5 and Figure 5.6 for details.

As can be seen in Figure 5.5, when α is set to be a small value (e.g. 0.001), there is sufficient randomness in tip selection random walk so that there are almost no differences for CTPS under weighted, unweighted and URTS strategies. Nevertheless, the α values have an obvious impact on CTPS in weighted random walk. As we know that larger α will increase the probability to select heavier tips so that most new coming transactions will be attached to the heaviest path eventually. This can well explain the CTPS decreasing trends as λ increases in Figure 5.5. On the other hand, different distances do not tell much difference as shown in Figure 5.6, which means that network delays have a limited influence on throughput under the examined situations.

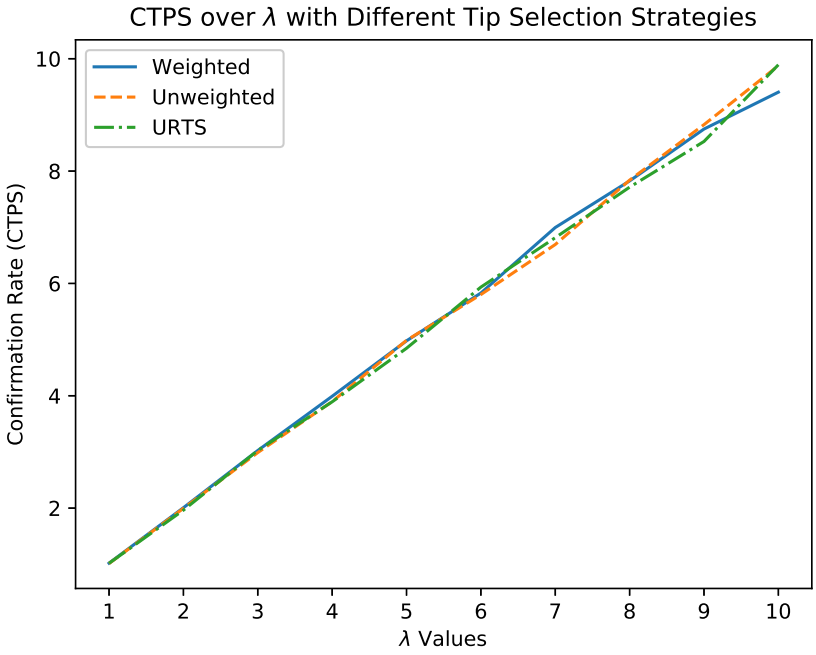

Figure 5.4: CTPS in Different Tip Selection Algorithms

Table 5.1: Experimental Environment; setup and configurations.

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

5.3.2 Simulation Model Validation

To validate the simulation model, we employ IOTA Implementation Reference[2] (IRI 1.6.1) with Docker to deploy a private network including 20 nodes on the SAVI OpenStack cloud platform[3]. Each node is a virtual machine with the flavor of medium size, see Table. 5.1 for configuration details. The open-source Compass[4] 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.

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

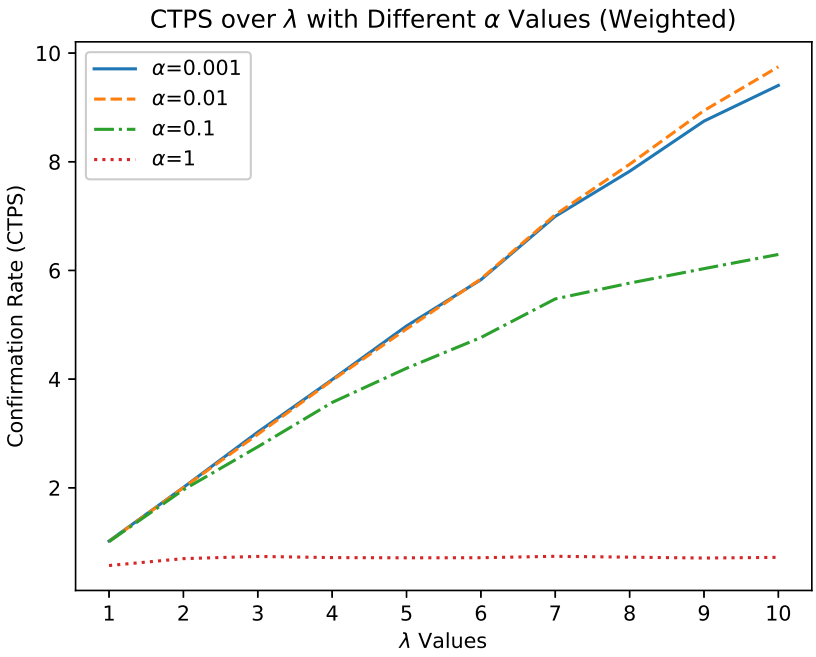

Figure 5.5: CTPS in Different α Values

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 configurations shown in Table. 5.1. We run all experiments with the transaction requests in a Poisson process, i.e., the transaction inter-arrival time follows an exponential distribution, with the total λ values varying from 1 to 10 with $step = 10$. In this experiment, the socket ZeroMQ[5] is used to listen and receive transaction data. In total, 39,462 confirmed transactions are collected from the experimental private IOTA network. To simplify the problem, we only send zero-value transactions so that every *Bundle* is composed with the transaction itself. Here, we present confirmed transactions statistics for all examined transaction arrival rates with the mean confirmed transactions of 10 milestones, as shown from Figure 5.7 to Figure 5.16.

By comparing the confirmations in experiments and simulations under dif-

[5]https://docs.iota.org/docs/iri/0.1/concepts/zero-message-queue

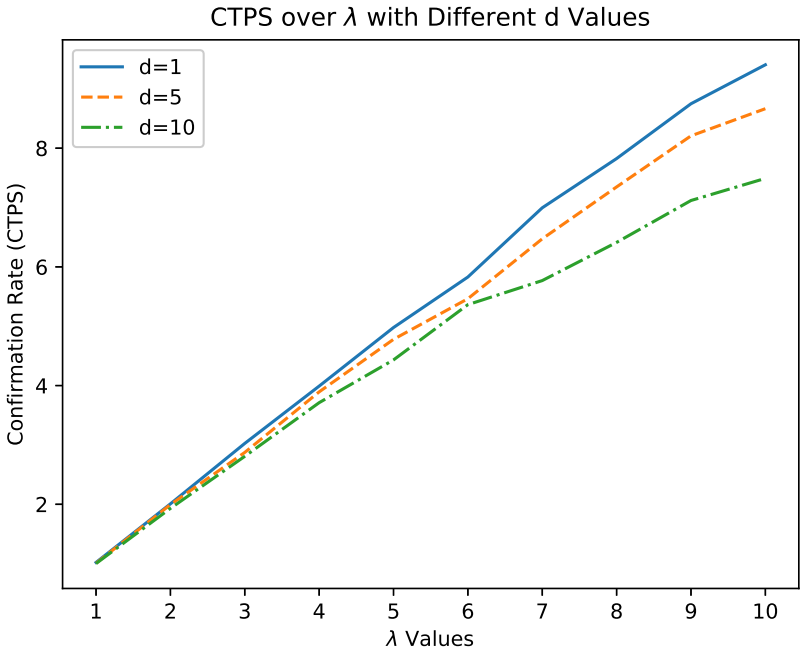

Figure 5.6: CTPS in Different Distancess

ferent λ values, we observe that the simulations results are very close to our experimental data in both mean confirmed transactions and confirmations by each milestone. Therefore, it is confident to use our extended simulator to conduct more simulations.

Then, we compare the experimental CTPS with the corresponding simulation data to examine the simulation model accuracy, as shown in Figure 5.17.

As can be seen in Figure 5.17, the simulation and experimental results are matched with an accuracy of more than 93%. This shows that our simulation model is effective in predicting CTPS in low λ situations.

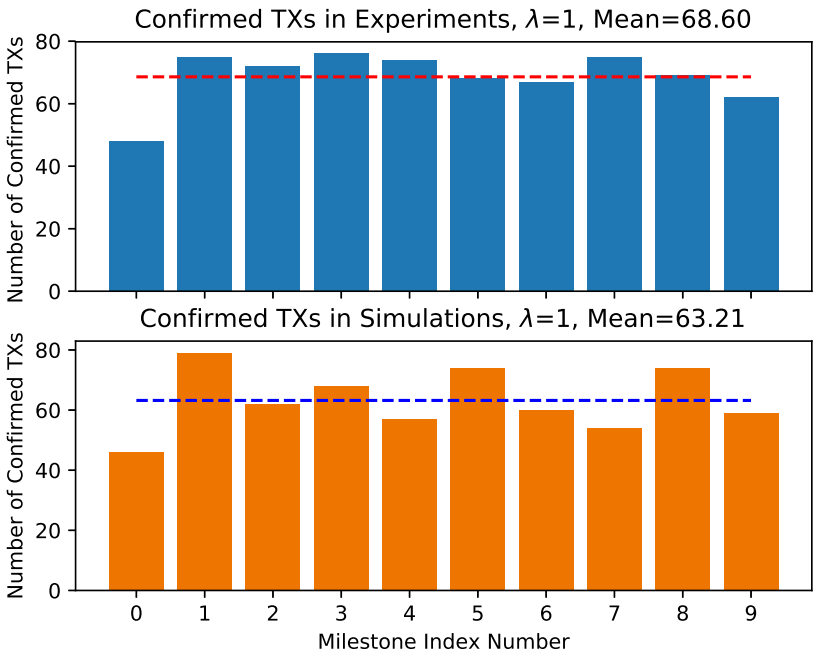

Figure 5.7: Experimental and Simulation Comparison, λ=1

5.4 Analytical Layered Model

The previous simulation modeling can provide a general relationship between CTPS and λ. However, it is difficult to describe more details such as how these confirmations are distributed in the Tangle. Therefore, we propose a layered model to explore the confirmation distributions in each single graph layer.

5.4.1 Layered Model

In the DAG of IOTA Tangle, we define a layer as all the confirmed transactions with the same depth from a Milestone in a hierarchical architecture, as shown in Figure 5.18. 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 5.18, 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

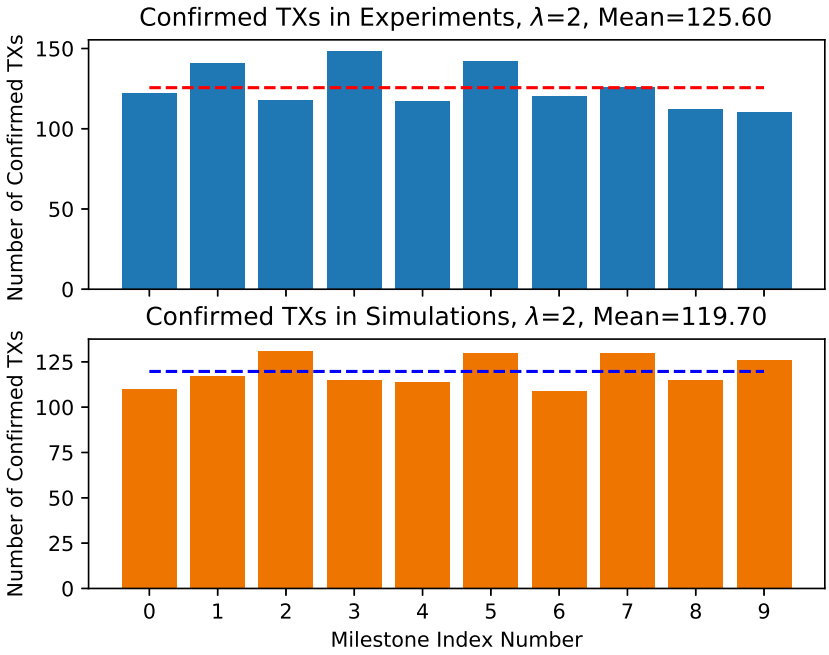

Figure 5.8: Experimental and Simulation Comparison, λ=2

$Layer_3$. With respect to this layering decomposition and using simulations data, we extract the transaction confirmations number in each layer of the DAG.

After plotting the confirmed transactions over layer number for each λ, we observe a lot of bell-shape curves, as shown from Figure 5.19 to Figure 5.28, which point us to the nonlinear models 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 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

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

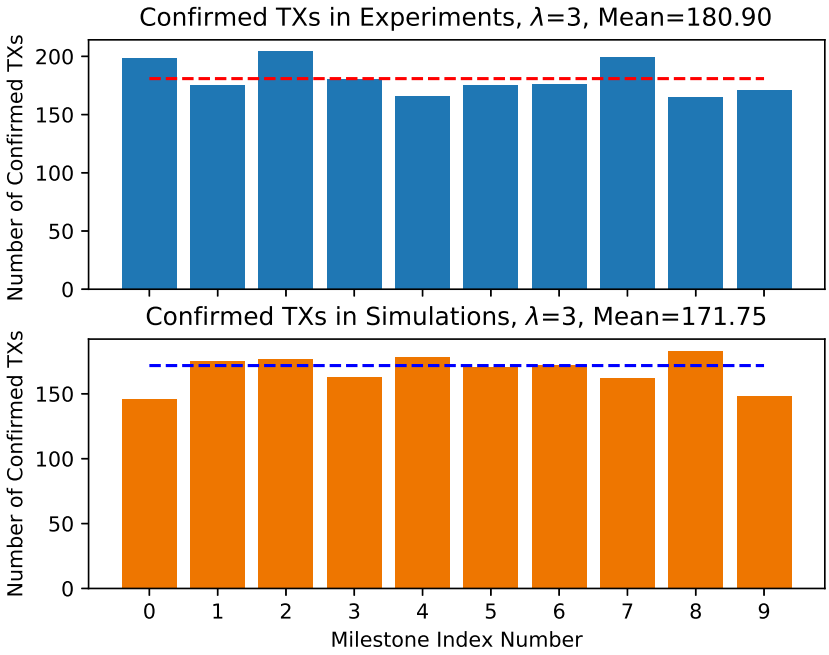

Figure 5.9: Experimental and Simulation Comparison, λ=3

Model outperforms others and is always listed in top 3 models, as we can see from the example of λ=10 in Figure 5.29. 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 5.2.

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. So, we have the number of confirmed transactions

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \tag{5.3}$$

Here, b has an almost linear increase trend as λ increases, while c almost remains the same from our simulation data in Table 5.2. This indicates that

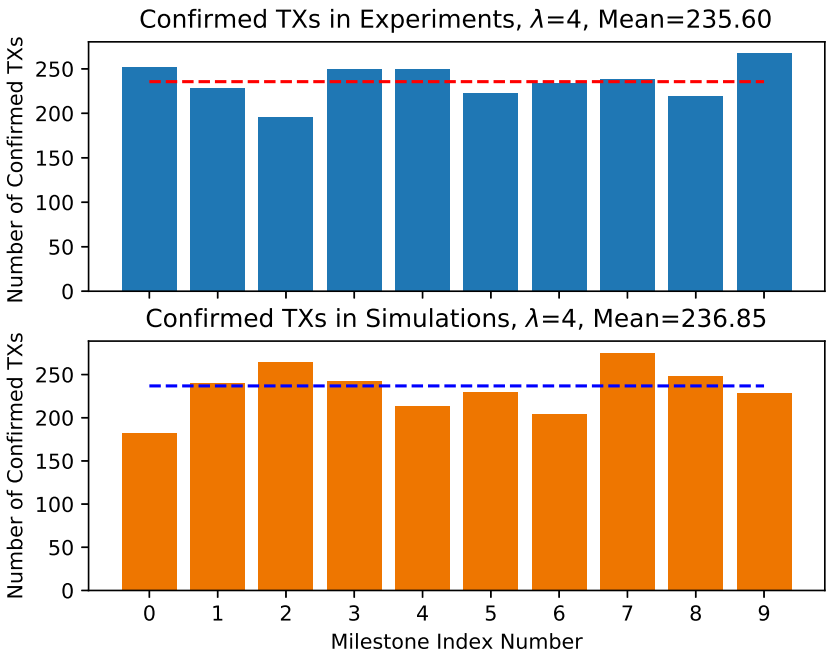

Figure 5.10: Experimental and Simulation Comparison, λ=4

all examined Gaussian Model has very similar shape; and the next random confirmed transaction is expected to be located at a deeper layer in the Tangle with higher λ values.

Moreover, the *Confidence Interval (CI)* of Gaussian models can be used to estimate the length of time to wait before reattaching transactions. However, let us first 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 estimation formula is as following; and the values are shown in Table 5.2 as CIUP.

Given that

$$Z = \frac{X - b}{c} \tag{5.4}$$

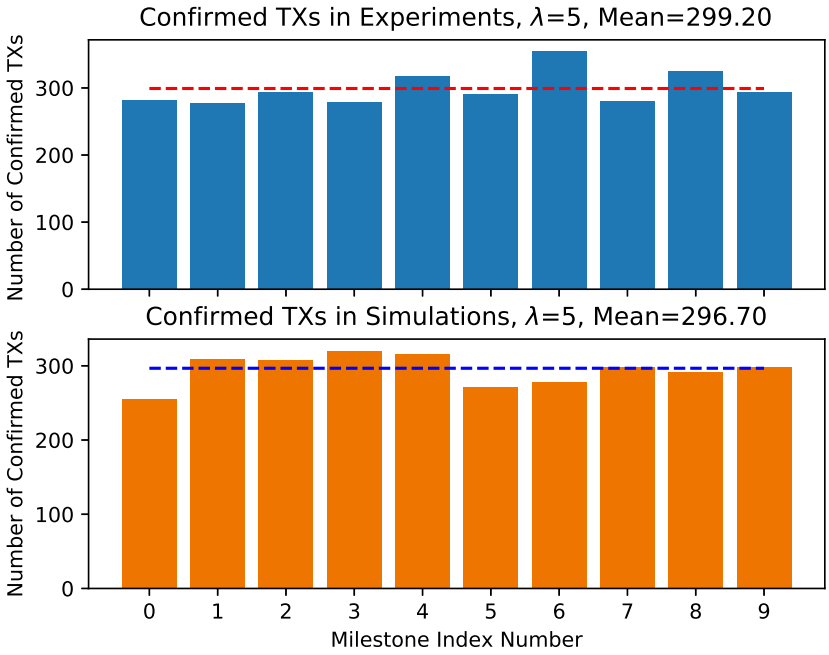

Figure 5.11: Experimental and Simulation Comparison, λ=5

So, we have

$$X_{\text{Upper}} = Zc + b \tag{5.5}$$

Then, we translate the *Upper Bound* layer to the time dimension by analyzing the layered model. As we notice from Figure 5.29, the decreasing happens just after a specific layer. From Figure 5.18, 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 5.18, 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 average waiting time before reattaching for users is estimated to be around $\mathbf{2}/\lambda_M$ seconds.

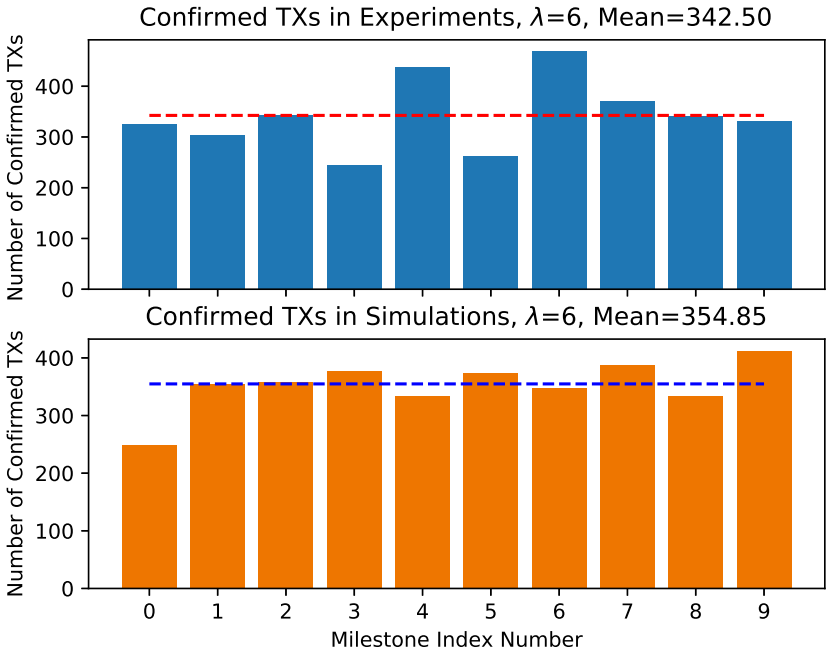

Figure 5.12: Experimental and Simulation Comparison, λ=6

5.4.2 Validation of Analytical Layered Model

To validate the deductive results of our layered model, we use the experimental data to conduct a statistical analysis. We find that 40,023 transactions in total are generated and sent to the network. Eventually, 39,462 of them are confirmed by milestones, within only 2,235 are confirmed after 2 minutes. As we know that the λ_M is set to be 1/60, i.e. a milestone is issued every minute, so $\mathbf{2}/\lambda_M$ seconds are right 2 minutes in our experiments. Therefore, we have 5.7% of the transactions confirmed after the time of $\mathbf{2}/\lambda_M$, which is relatively low and is well matched with the prediction of our layered model.

5.5 Summary

In this paper, we studied the performance of private IOTA network by experimental, simulation and analytical modeling. We leveraged these models to answer two research questions on throughput and RWT, with high level

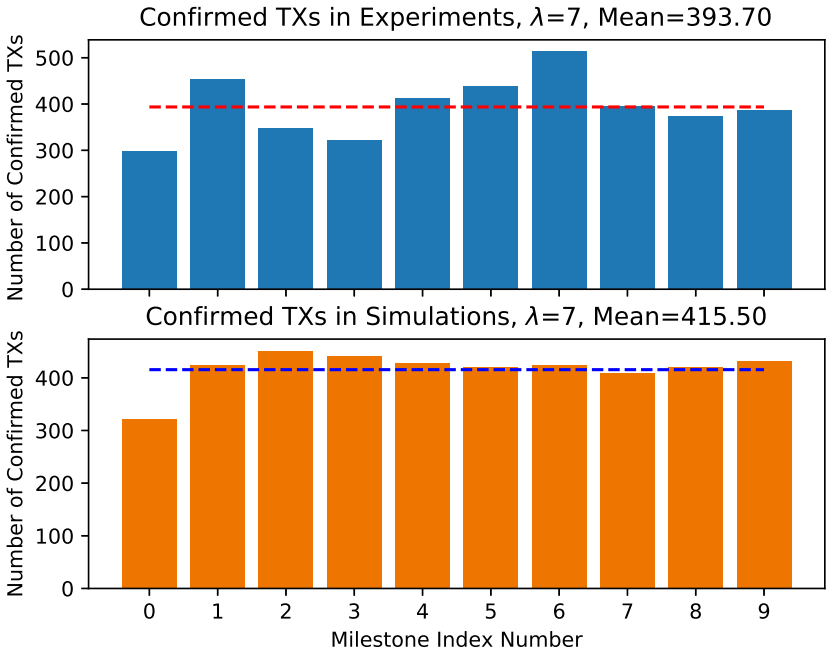

Figure 5.13: Experimental and Simulation Comparison, λ=7

of confidence. Through the Simulation Model, we empirically explored and analyzed the influences of transaction arrival rate λ, *Tip Selection Algorithm*, randomness α in weighted random walk algorithm and network delay D on CTPS. Among all these impact factors, we found that λ was the most important one, which has an almost linear relationship with the CTPS. Moreover, we leveraged the proposed analytical layered model to explore the confirmation distributions and found that the confirmations are normally distributed in DAG layers, which led to characterizing the Gaussian Model. Using this model, we also estimated the RWT for a private IOTA network which was validated by our experimental results. In conclusion, our proposed performance models provided important insight on the performance of IOTA distributed ledger.

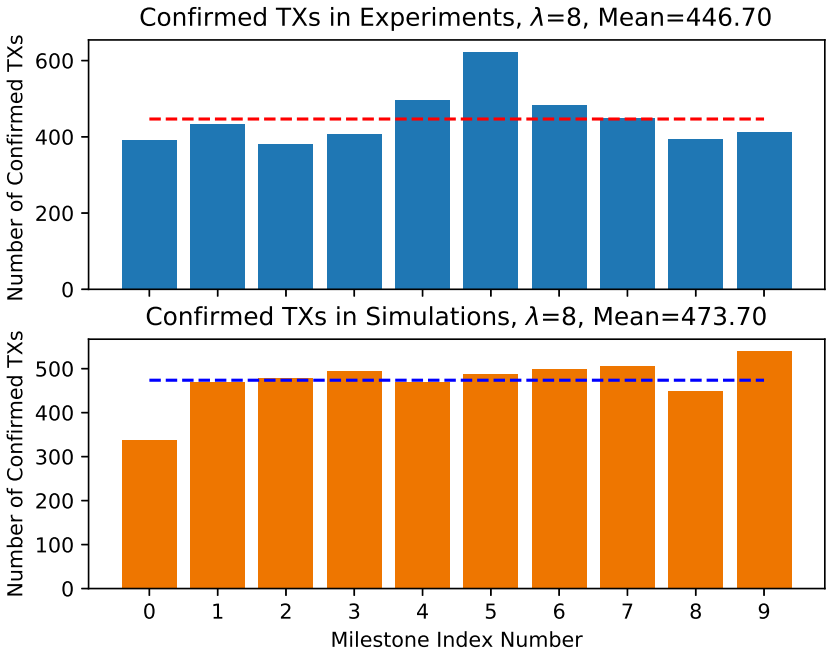

Figure 5.14: Experimental and Simulation Comparison, λ=8

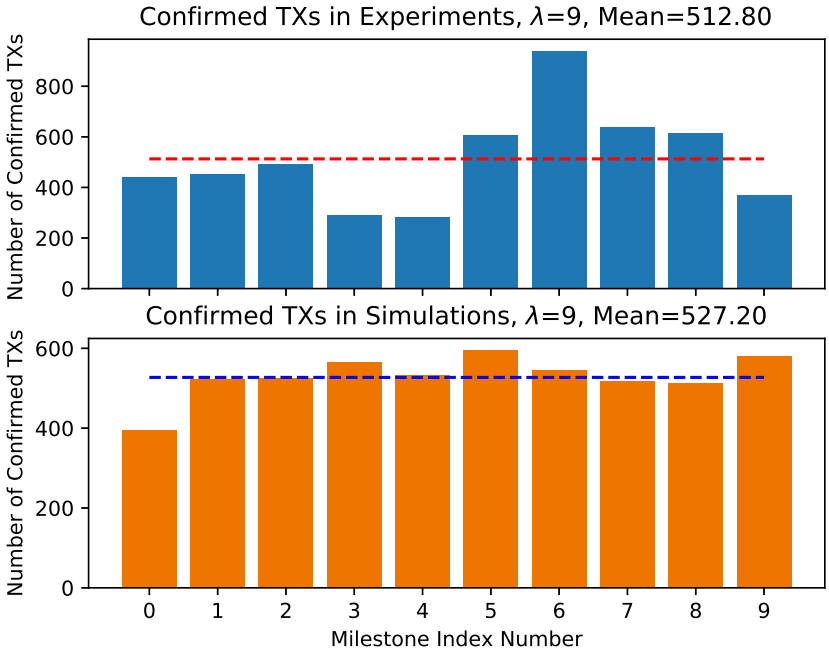

Figure 5.15: Experimental and Simulation Comparison, λ=9

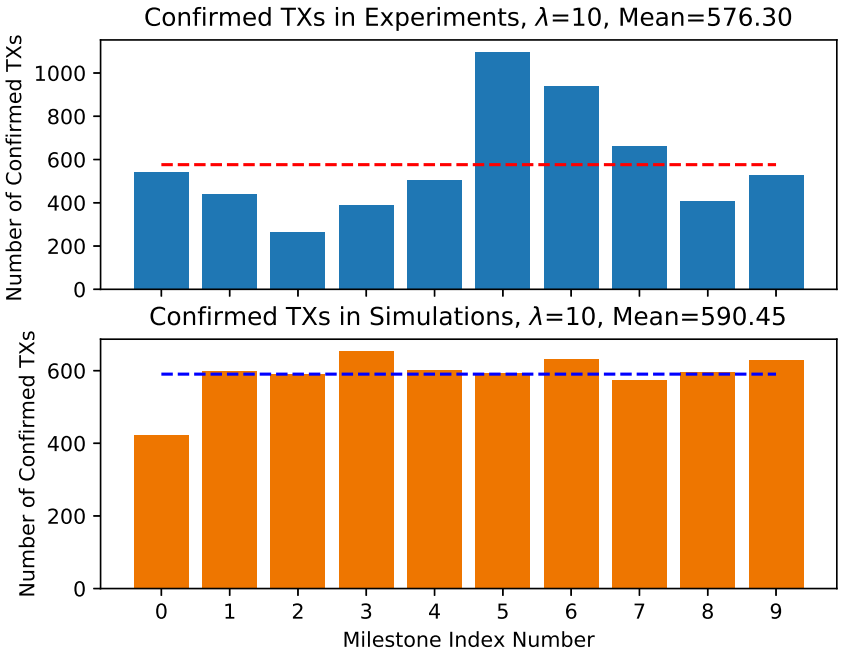

Figure 5.16: Experimental and Simulation Comparison, λ=10

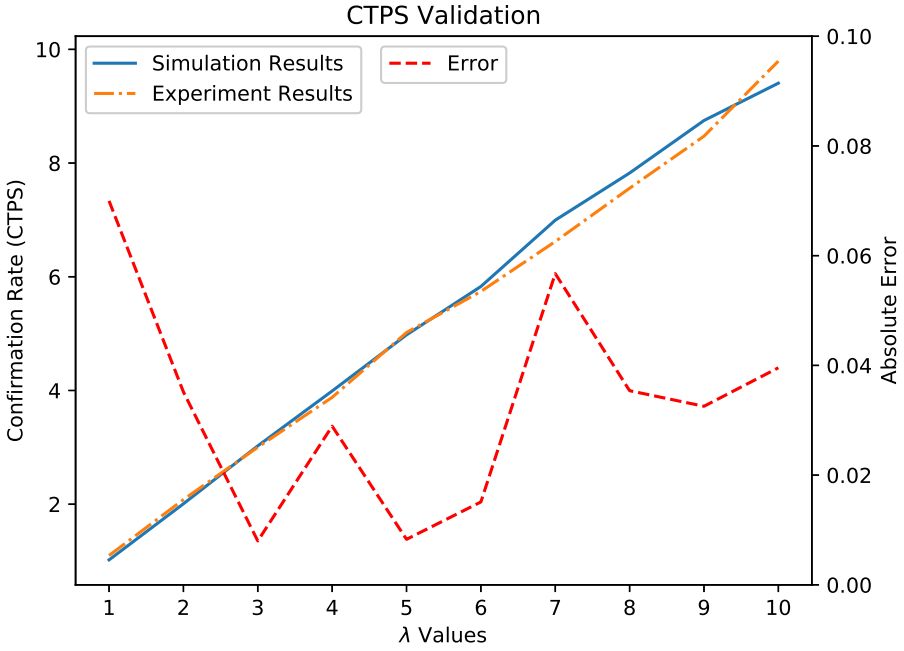

Figure 5.17: Experimental and Simulation CTPS Comparison

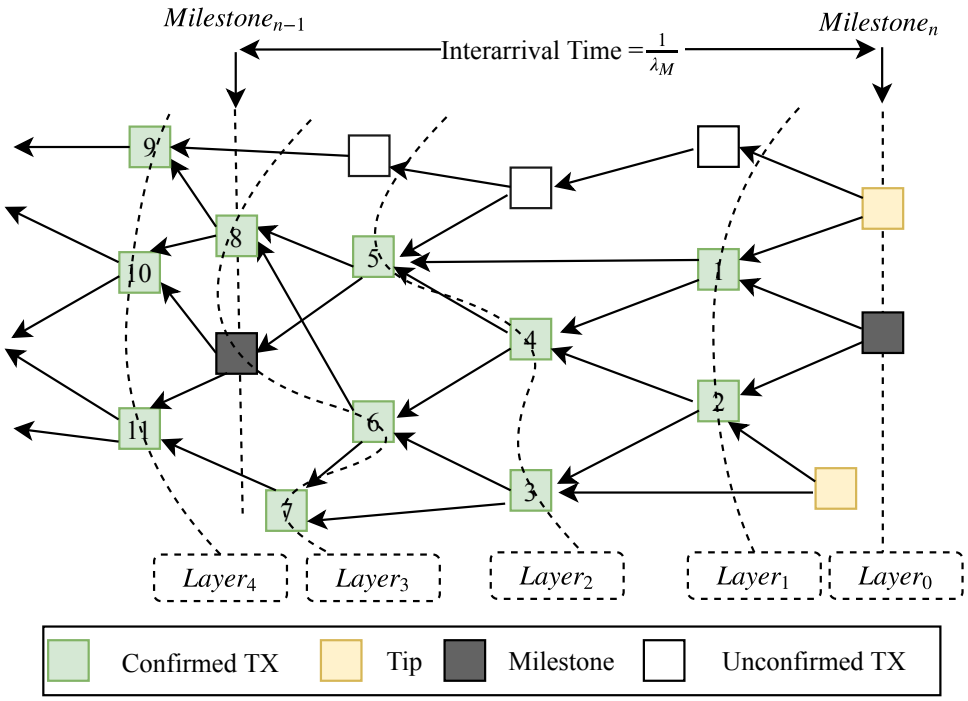

Figure 5.18: Layered Model for Transaction Confirmations

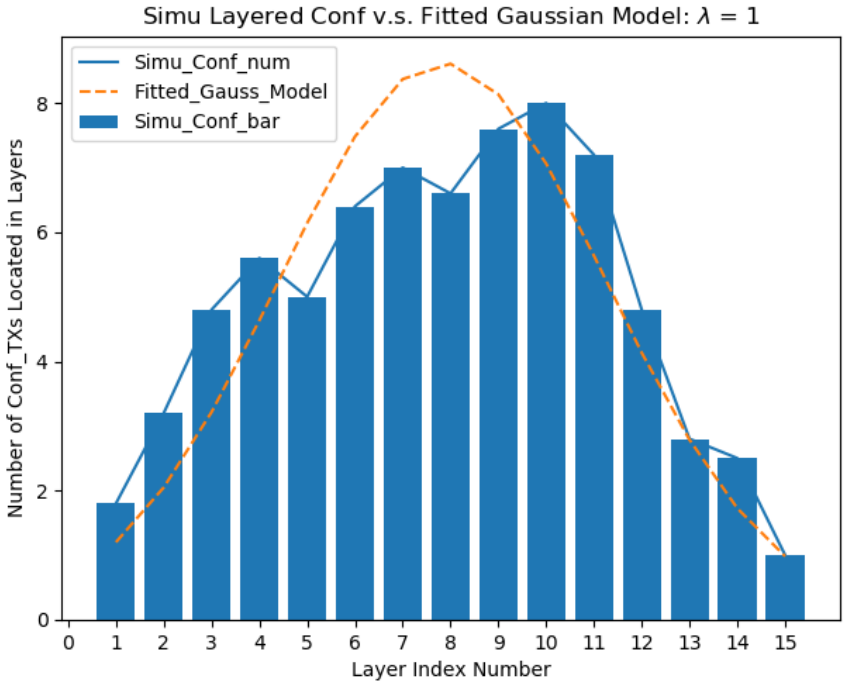

Figure 5.19: Layered Confirmed Transactions Bell-shape for λ=1

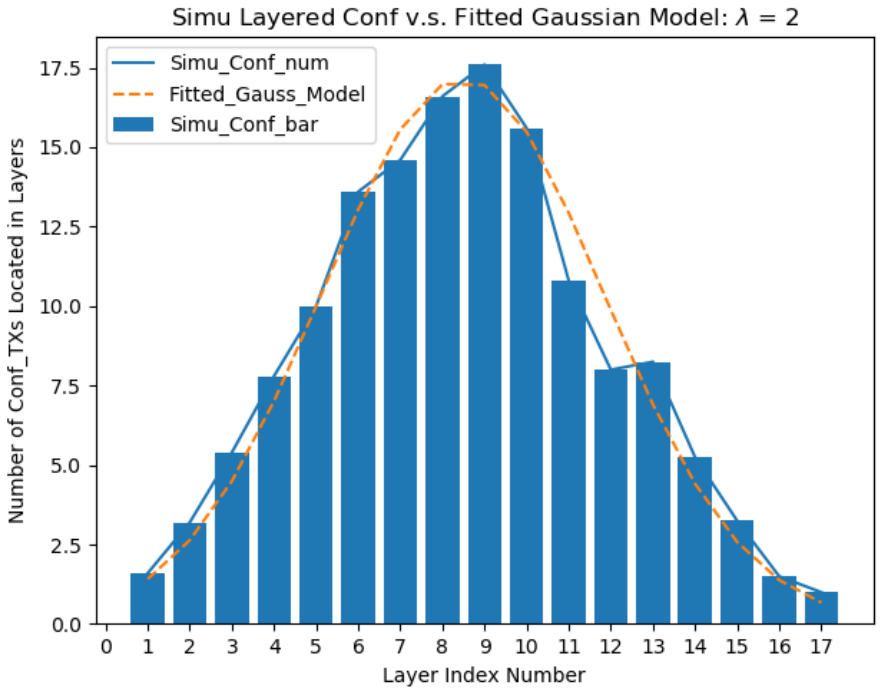

Figure 5.20: Layered Confirmed Transactions Bell-shape for λ=2

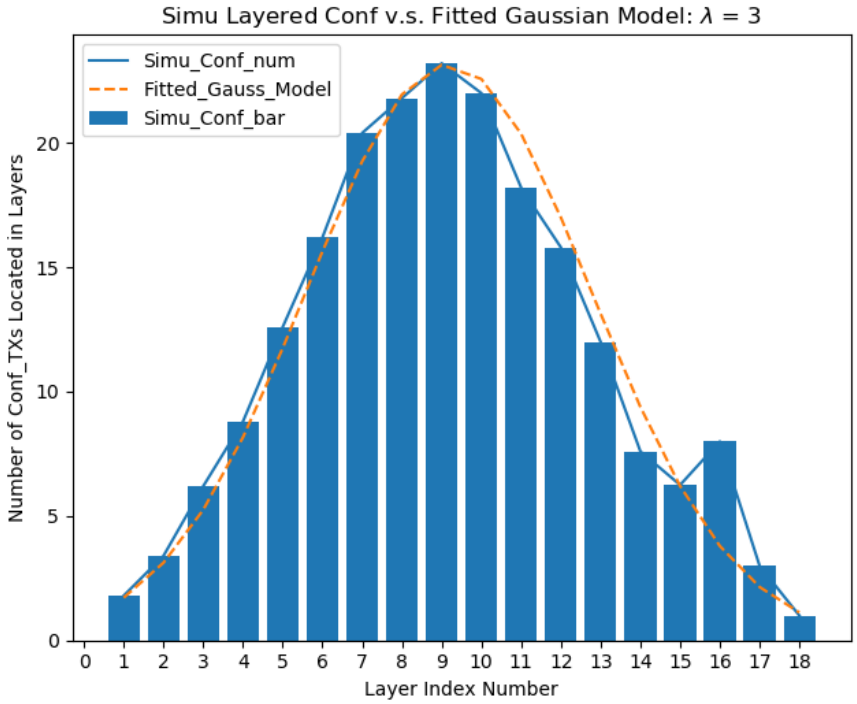

Figure 5.21: Layered Confirmed Transactions Bell-shape for λ=3

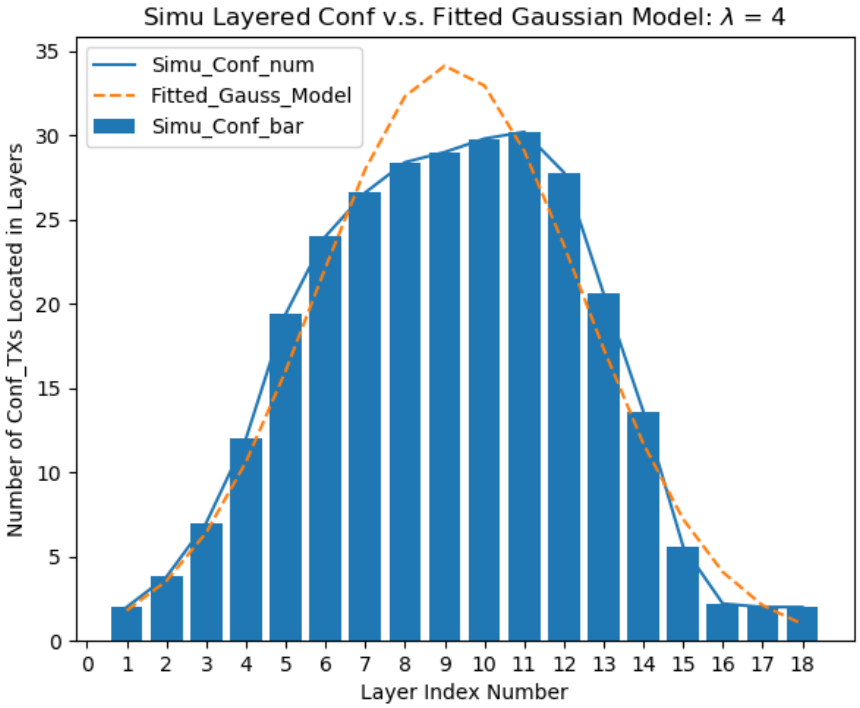

Figure 5.22: Layered Confirmed Transactions Bell-shape for λ=4

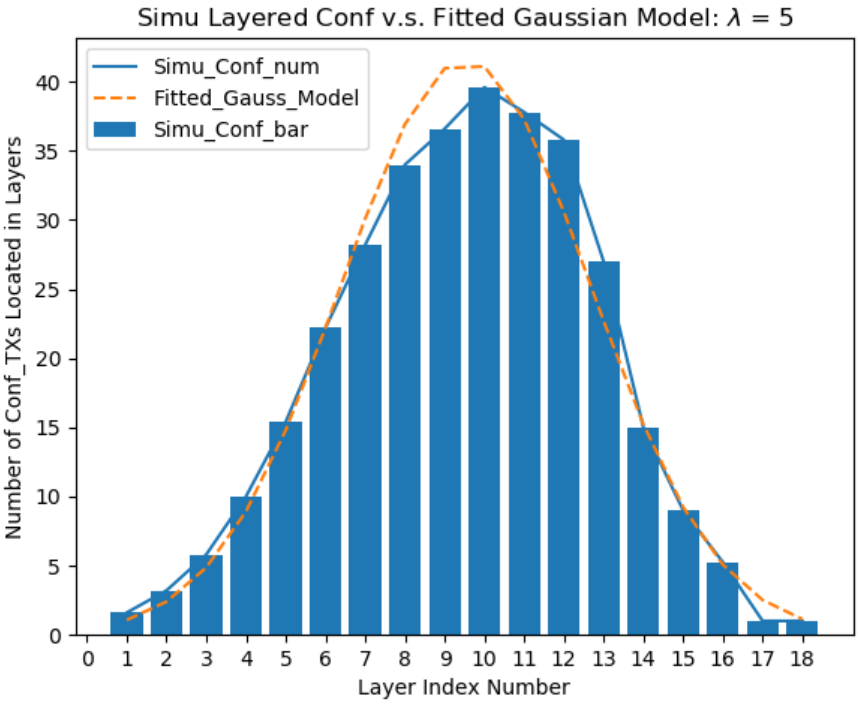

Figure 5.23: Layered Confirmed Transactions Bell-shape for λ=5

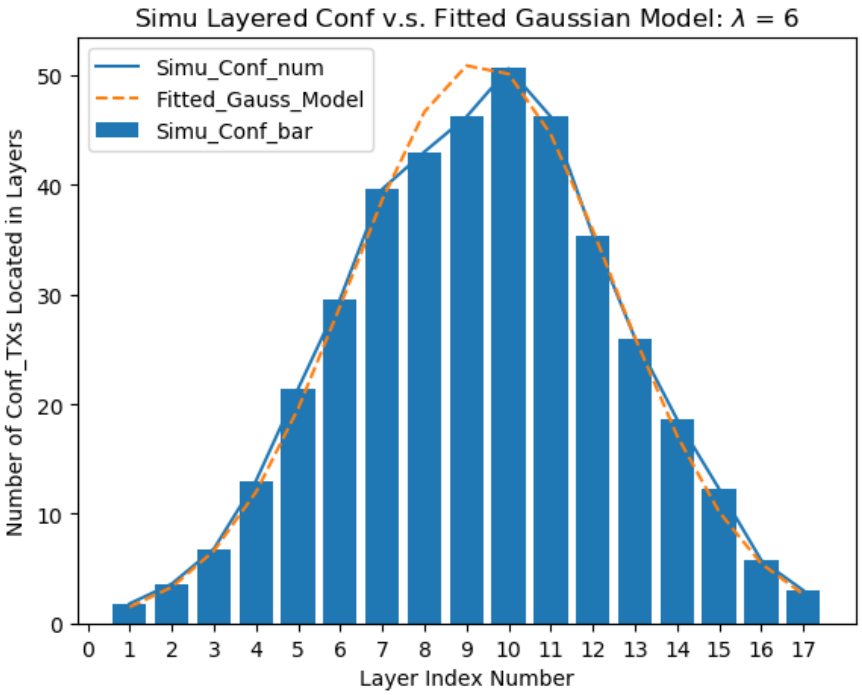

Figure 5.24: Layered Confirmed Transactions Bell-shape for λ=6

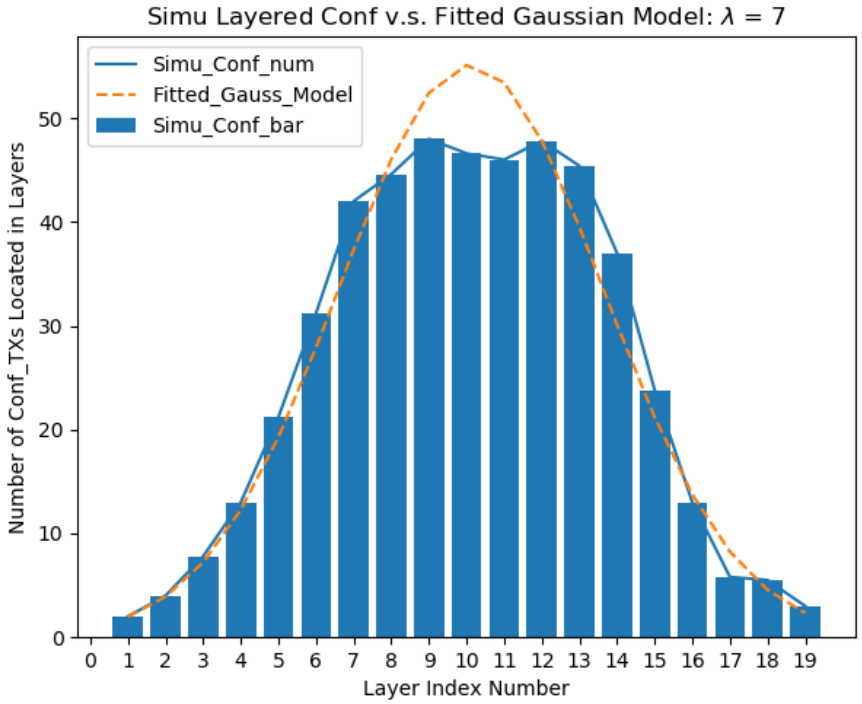

Figure 5.25: Layered Confirmed Transactions Bell-shape for λ=7

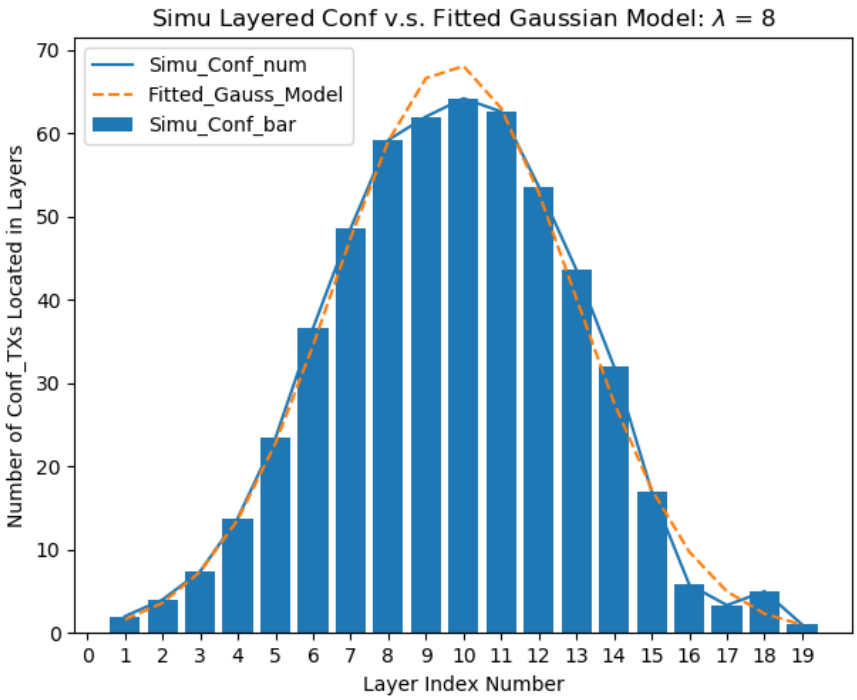

Figure 5.26: Layered Confirmed Transactions Bell-shape for λ=8

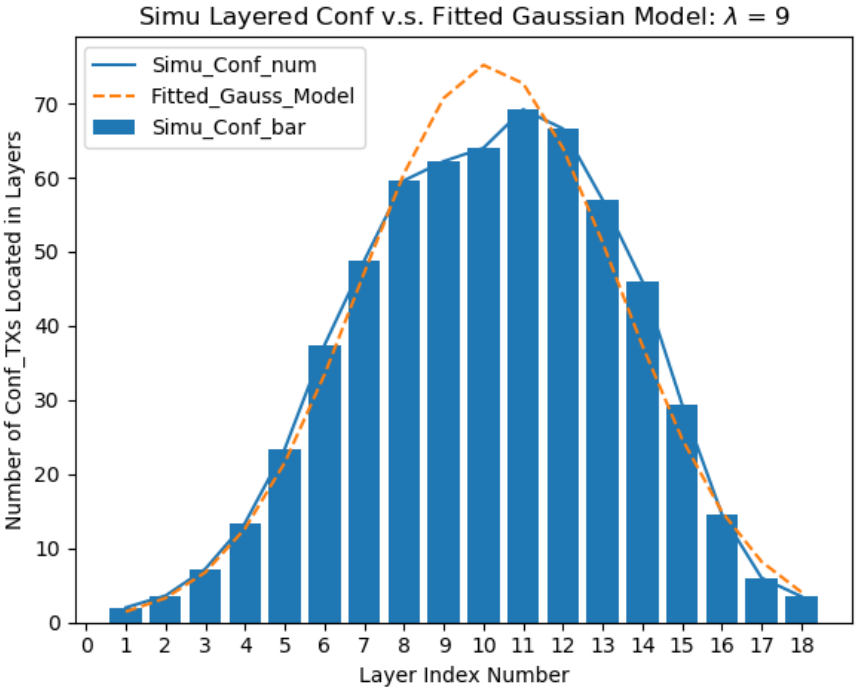

Figure 5.27: Layered Confirmed Transactions Bell-shape for λ=9

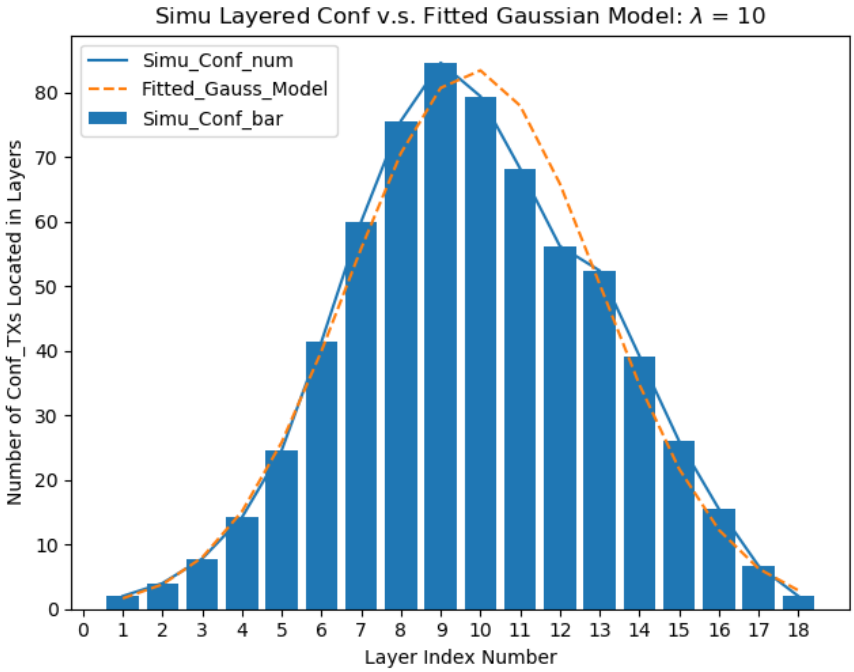

Figure 5.28: Layered Confirmed Transactions Bell-shape for λ=10

Table 5.2: Model Fitting Results for Different λ Values

Statistics of Gaussian Model										
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	7.60	16.56	22.44	32.09	40.23	49.38	51.80	66.69	71.02	81.01
b	7.94	8.36	8.98	9.17	9.73	9.39	10.17	9.75	10.30	9.72
c	4.25	3.517	3.74	3.66	3.30	3.30	3.86	3.30	3.54	3.28
AMUP*	13.60	15.00	15.60	16.80	16.40	16.20	17.60	17.00	17.40	16.60
CIUP+	16.27	15.25	16.31	16.34	16.19	15.85	17.73	16.21	17.23	16.16

*Actual Mean Upper Bound, +Confidence Interval Upper Bound

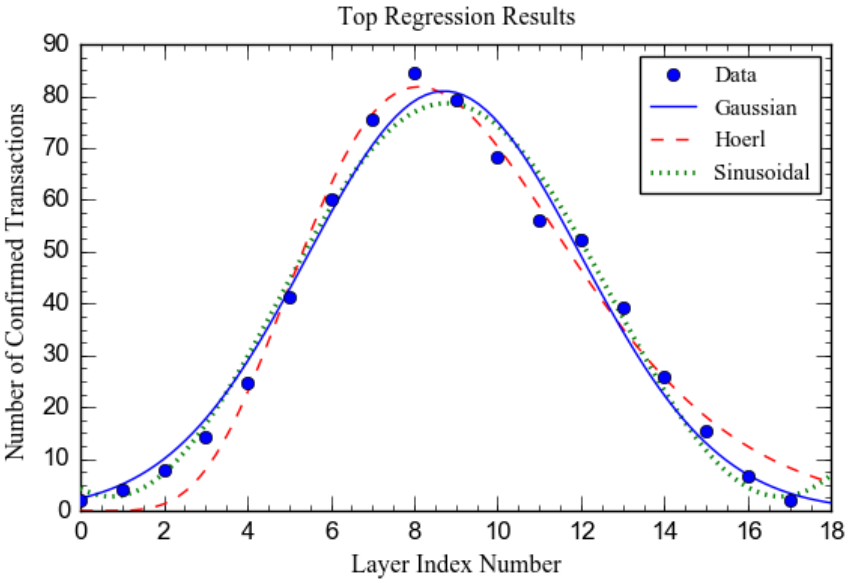

Figure 5.29: Simulation Data and Fitted Models for λ=10

Chapter 6

Discussion

6.1 System Security

In IOTA protocol, all peers in the network are responsible for transaction validation and system security to protect attacks such as double-spending and Sybil attacks. Both can be conducted by attackers through issuing large amount of transactions in a short time interval. To protect these attacks, on one hand, user can increase the difficulty of proof of work in large transactions to increase the difficulty of dominating the network through hashpower; on the other hand, IOTA employs a powerful Coordinator to continuously generate milestones which are acting as honest and trusted transactions. This is very important when IOTA is in its infancy stage. As we know that the number of assiduous honest transactions must always be found to be in the majority.

In performance analysis, reattachment is used to improve the confirmation probability and speed. This may raise a question of double-spending if the two identical transactions are both confirmed and added to the ledger. Will this situation really happen? Let's discuss the double-spending problem in Figure 3.6. For example, if a user firstly issues transaction v_5, and in a very short time reattaches the same transaction as v_7 to another position, these are two conflicting transactions added by a user in different areas of the tangle. When transactions v_8 and v_9 come to the Tangle, they will not see the conflict

and validate their chosen tips, i.e., (v_5, v_9) and (v_7, v_9), respectively.

However, in both COO or COO-less consensus, there always exists a propagation delay for transactions to be attached to the Tangle and get confirmed. Therefore, sooner or later it must be the case that both conflicting transactions are in the path of validation of one transaction during the delay. For example, when transaction X is added to the Tangle, it will detect the conflict when validating and is not attach to the elected tips. Instead, in order to be a valid transaction, tip X will re-select tips until no conflicting ones are selected. The same strategies are applied to other new coming tips, so that one of the conflicting transactions will be abandoned eventually. Technically, the double-spending protection is achieved this way.

6.2 Simulator Efficiency

As we can see from both experimental and simulation results, the transaction confirmation rate has a similar linear relationship over examined low arrival rates. However, it is difficult to explore the confirmation rate under large scale situations e.g. big arrival rates with enough milestones, due to the experimental hashpower limitation and simulation efficiency bottleneck. For example, it took about 3.6 hours to simulate MCMC weighted random walk IOTA protocol with $transactions = 6000, agents = 20, d = 1$ and $\alpha = 0.001$ in our environment; we ran the simulation on a DELL PC with Windows 10 OS, 8^{th} generation Intel core$^{\text{TM}}$ i7-8700 12-Core processor and 16GB RAM. As M. Zander et al. stated, the cumulative weights updating process mainly contributed to the running time, with a ratio 61.81% [56]. Additionally, we found that the tip selection random walk always started from the genesis transaction, which required updating weighted for all transactions. However, our Layered Model shows that there are almost not any confirmations after the layer reaches to

an Upper Bound, e.g. 20. This finding is well matched with the empirical analyses of B. Kusmierz [3], who concluded that any starting position placed further than 10λ to 20λ provided the same growth of number of tips, and it would not influence the tip number. Therefore, we safely state that it is not necessary to start from the Genesis for each random walk. This can be used to improve simulator efficiency, by thinking the old transactions as a "Black hole" which begins with the genesis.

Chapter 7

Conclusion and Future Work

In this work, we have explored the applications of distributed ledger technologies, e.g. IOTA, in IoT. Basically, we strove to find out answers for the questions on how can we use DLTs to build IoT systems and how is the system scalability and performance for such scenarios. More specifically, we have first proposed a DAG-based distributed ledger solution for smart communities to handle inter-house distributed energy resources transactions. With the initial experimental results, our solution provides high TPS/CTPS and good scalability which is important for IoT applications. We conclude that our proposed DAG-based distributed ledger is an effective solution for building a smart home IoT infrastructure.

Then, we studied the performance of private IOTA network by experimental, simulation and analytical modeling. We leveraged these models to answer two research questions on throughput and RWT, with high level of confidence. Through the Simulation Model, we empirically explored and analyzed the influences of transaction arrival rate λ, tip selection algorithm, randomness α in weighted random walk algorithm and network delay D on CTPS. Among all these impact factors, we found that λ was the most important one, which has an almost linear relationship with the CTPS. Moreover, we leveraged the pro-

posed analytical layered model to explore the confirmation distributions and found that the confirmations are normally distributed in DAG layers, which led to characterizing the Gaussian Model. Using this model, we also estimated the RWT for a private IOTA network which was validated by our experimental results. In conclusion, our proposed performance models provided important insight on the performance of IOTA distributed ledger.

As for the future work, one research direction could be to optimize the designed system by removing COOs and only employing MCMC for consensus to achieve better decentralization. Next, it is significant to explore the average transaction confirmation time which denotes the transaction time latency. More work on the comparisons with other IoT-oriented DL consensuses should also be conducted. For performance analysis, on one hand, one possible direction could be to quantitatively study how other factors influence the confirmation rate such as network scale, network delay and the value of α. It is also interesting to research IOTA performance under COO-less consensus in further study. On the other hand, improving the DAGSim to make it suitable for running simulation scenarios at large scale would be another direction of our future research.

References

[1] J. Ahmed, "OruMesh White Paper," *Whitepaper*, pp. 1–13, 2017. [Online]. Available: https://orumesh.com/whitepaper2.0.pdf. 19, 20

[2] M. Alharby and A. Van Moorsel, "BlockSim: A Simulation Framework for Blockchain Systems," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 135–138, 2018, ISSN: 01635999. DOI: 10.1145/3308897.3308956. 13, 15

[3] B. Kusmierz, "The first glance at the simulation of the Tangle : discrete model," *IOTA Found. WhitePaper*, pp. 1–10, 2017. [Online]. Available: https://iota.org/simulation%7B%5C_%7Dtangle-preview.pdf. 14, 31, 78

[4] M. Bottone, F. Raimondi, and G. Primiero, "Multi-agent based simulations of block-free distributed ledgers," in *Proc. - 32nd IEEE Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2018*, vol. 2018-Janua, 2018, pp. 585–590, ISBN: 9781538653944. DOI: 10.1109/WAINA.2018.00149. 15

[5] CBinsights, "Banking Is Only The Beginning: 42 Big Industries Blockchain Could Transform," CBinsights, Tech. Rep., 2018. [Online]. Available: https://www.cbinsights.com/research/industries-disrupted-blockchain/. 2

[6] D. Cheatoshin, *The Dagger crypto currency: white paper*. [Online]. Available: https://github.com/XDagger/xdag/blob/master/WhitePaper.md. 20

[7] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016, ISSN: 21693536. DOI: 10.1109/ACCESS.2016.2566339. 10, 11

[8] A. Churyumov, "Byteball : A Decentralized System for Storage and Transfer of Value," Tech. Rep., 2017, pp. 1–49. [Online]. Available: https://byteball.org/Byteball.pdf. 19

[9] *Compass overview*. [Online]. Available: https://docs.iota.org/docs/compass/0.1/introduction/overview. 22

[10] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," *2016 IEEE/ACS 13th Int. Conf. Comput. Syst. Appl.*, pp. 1–6, 2016, ISSN: 21615330. DOI: 10.1109/AICCSA.2016.7945805. [Online]. Available: http://ieeexplore.ieee.org/document/7945805/. 10

[11] Digiconomist, *Bitcoin Energy Consumption Index*, 2018. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption. 3

[12] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *Proc. Second Int. Conf. Internet-of-Things Des. Implement. - IoTDI 2017*, 2017, pp. 173–178, ISBN: 9781450349666. DOI: 10.1145/3054977.3055003. arXiv: 1602.05561. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3054977.3055003. 1, 2

[13] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE Int. Conf. Pervasive Comput. Commun. Work. (PerCom Work.*, pp. 618–623, 2017, ISSN: 2474-2503. DOI: 10.1109/PERCOMW.2017.7917634. [Online]. Available: http://ieeexplore.ieee.org/document/7917634/. 11

[14] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy," *arXiv Prepr. arXiv1712.02969*, pp. 1–17, 2017. arXiv: 1712.02969. [Online]. Available: http://arxiv.org/abs/1712.02969. 11

[15] C. Fan, H. Khazaei, Y. Chen, and P. Musilek, "Towards A Scalable DAG-based Distributed Ledger for Smart Communities (in press)," in *IEEE 5th World Forum Internet Things*, Limerick, 2019. [Online]. Available: https://www.researchgate.net/publication/331485141%7B%5C_%7DTowards%7B%5C_%7DA%7B%5C_%7DScalable%7B%5C_%7DDAG-based%7B%5C_%7DDistributed%7B%5C_%7DLedger%7B%5C_%7Dfor%7B%5C_%7DSmart%7B%5C_%7DCommunities. 12

[16] C. G., *5 Types of Blockchain Consensus Mechanisms*, 2018. [Online]. Available: https://www.logicsolutions.com/5-types-blockchain-consensus-mechanisms/. 2

[17] A. Gal, *The Tangle: an Illustrated Introduction*, 2018. [Online]. Available: https://blog.iota.org/the-tangle-an-illustrated-introduction-4d5eae6fe8d4. 15

[18] K. Hamzeh, J. Misic, and V. B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems," *IEEE Trans. PARALLEL Distrib. Syst.*, vol. 23, no. 5, pp. 936–943, 2012. DOI: 10.1109/TPDS.2011.199. 12

[19] E. Hop, *Exploring the IOTA signing process*, 2018. [Online]. Available: https://medium.com/iota-demystified/exploring-the-iota-signing-process-eb142c839d7f. 27, 28

[20] G. Hummer, *On sharding blockchains*, 2017. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-FAQs. 35

[21] Hyperledger, *Membership Service Providers (MSP)*, 2017. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.2/msp.html. 7, 39

[22] S. D. S. U. C. for International Communications and C. D. of Transportation, *Smart Communities Guidebook: How California's Communities Can Thrive in the Digital Age*. California Department of Transportation, 1997. [Online]. Available: https://books.google.ca/books?id=GNgAHAAACAAJ. 34

[23] IOTA Foundation, *Consensus on the Tangle*, 2018. [Online]. Available: https://docs.iota.org/introduction/tangle/consensus. 37

[24] H. Khazaei, M. Fokaefs, S. Zareian, Nasim Beigi-Mohammadi, Brian Ramprasad, M. Shtern, P. Gaikwad, and M. Litoiu, "How do I choose the right NoSQL solution? A comprehensive theoretical and experimental survey," *Big Data Inf. Anal.*, vol. 2, no. 1, 2016. 35

[25] B. Kusmierzkusmierz and P. Staupe, "Extracting Tangle Properties in Continuous Time via Large-Scale Simulations," *IOTA Found. WhitePaper*, 2018. 14, 32

[26] M. Lathif, P. Nasirifard, and H.-A. Jacobsen, "Demo abstract: Cidds: A configurable and distributed dag-based distributed ledger simulation framework," in *Middlew. 2018 - Proc. 2018 ACM/IFIP/USENIX Middlew. Conf.*, 2018, pp. 7–8, ISBN: 9781450361095. DOI: 10.1145/3284014.3284018. 15

[27] R. Lea, "Smart Cities: An Overview of the Technology Trends Driving Smart Cities," *Ieee*, vol. 3, no. March, pp. 1–16, 2017. 34

[28] C. LeMahieu, "Nano: A Feeless Distributed Cryptocurrency Network," *Whitepaper*, pp. 1–8, 2014. [Online]. Available: https://nano.org/en/whitepaper. 20

[29] C. Lemahieu, "Nano: A Feeless Distributed Cryptocurrency Network," *White Pap.*, pp. 1–8, 2018. 20, 21

[30] S. D. Lerner, "DagCoin Draft," Tech. Rep., 2015, pp. 1–6. [Online]. Available: https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf. 19

[31] Q. Li, J.-Y. Ma, and Y. Chang, "Blockchain Queue Theory," *arXiv:1808.01795v2*, 2018. arXiv: arXiv:1808.01795v2. [Online]. Available: https://arxiv.org/abs/1808.01795. 17

[32] M. A. Marsan, "Stochastic Petri nets: An elementary introduction," in *Adv. Petri Nets 1989. APN 1988. Lect. Notes Comput. Sci.* Berlin: Springer, 1990, pp. 1–29. DOI: 10.1007/3-540-52494-0_23. [Online]. Available: http://link.springer.com/10.1007/3-540-52494-0%7B%5C_%7D23. 12

[33] R. Memon, J. Li, and J. Ahmed, "Simulation Model for Blockchain Systems Using Queuing Theory," *Electronics*, vol. 8, no. 2, p. 234, 2019, ISSN: 2079-9292. DOI: 10.3390/electronics8020234. [Online]. Available: http://www.mdpi.com/2079-9292/8/2/234. 18

[34] H.-B. Mor, *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press (Feb. 18 2013), 2013. 17

[35] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *J. Gen. Philos. Sci.*, vol. 39, no. 1, pp. 53–67, 2008, ISSN: 09254560. DOI: 10.1007/s10838-008-9062-0. arXiv: 43543534534v343453. [Online]. Available: https://bitcoin.org/bitcoin.pdf. 2

[36] *NANO FAQ*. [Online]. Available: https://nano.org/en/faq. 20

[37] *NetLogo*. [Online]. Available: https://ccl.northwestern.edu/netlogo/. 15

[38] O. Novo, "Blockchain Meets IoT: an Architecture for Scalable Access Management in IoT," *IEEE Internet Things J.*, vol. 14, no. 8, pp. 1–12, 2018, ISSN: 23274662. DOI: 10.1109/JIOT.2018.2812239. 11

[39] H. Pervez, M. Muneeb, M. U. Irfan, and I. Ul Haq, "A Comparative Analysis of DAG-Based Blockchain Architectures," in *ICOSST 2018 - 2018 Int. Conf. Open Source Syst. Technol. Proc.*, IEEE, 2019, pp. 27–34, ISBN: 9781538695647. DOI: 10.1109/ICOSST.2018.8632193. 18, 19

[40] S. Popov, "The Tangle," *New Yorker*, 2018, ISSN: 0028-792X. [Online]. Available: http://www.vanderbilt.edu/viibre/members/documents/12960-Weiner-NY-2005.pdf. 1, 3, 7, 14, 15, 18, 19, 48

[41] P. B. Pureswaran and V., "Device democracy: Saving the future of the Internet of Things," New York, NY, USA, Tech. Rep., 2014, p. 4. [Online]. Available: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=XB%7B%5C&%7Dinfotype=PM%7B%5C&%7Dappname=GBSE%7B%5C_%7DGB%7B%5C_%7DTI%7B%5C_%7DUSEN%7B%5C&%7Dhtmlfid=GBE03620USEN%7B%5C&%7Dattachment=GBE03620USEN.PDF. 1

[42] R. Radhakrishnan and B. Krishnamachari, "Streaming Data Payment Protocol (SDPP) for the Internet of Things," Los Angeles, 2018. 12

[43] Y. Ribero and D. Raissar, "Dagcoin whitepaper," *Whitepaper*, no. May, pp. 1–71, 2018. [Online]. Available: https://dagcoin.org/whitepaper.pdf. 19

[44] R. Saulo, Z. Artur, F. Eduardo, S. Jose Eduardo, M. D. Sadoc, and A. Borges Vieira, "Learning Blockchain Delays : A Queueing Theory Approach," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 122–125, 2018. 17

[45] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Serialization: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections," *White Pap.*, pp. 1–66, 2017. DOI: 10.4135/9781483340333NV-3. [Online]. Available: https://pdfs.semanticscholar.org/a45d/5b355f5181dcd700ec5e129d8b6e195d0be1.pdf?%7B%5C_%7Dga=2.249570438.164899734.1525105958-2122034482.1513872018. 21

[46] Y. Sompolinsky, A. Zohar, and C. Science, "Phantom , Ghostdag : Two Scalable BlockDAG protocols," *White Pap.*, pp. 1–17, 2018. 21

[47] D. Strugar, R. Hussain, M. Mazzara, V. Rivera, J. Lee, and R. Mustafin, "On M2M Micropayments : A Case Study of Electric Autonomous Vehicles," *arXiv Prepr. arXiv1804.08964*, pp. 2–5, 2018. arXiv: 1804.08964. [Online]. Available: http://arxiv.org/abs/1804.08964. 12

[48] H. Sukhwani, "Performance Modeling & Analysis of Hyperledger Fabric (Permissioned Blockchain Network)," Doctor of Philosophy Thesis, Duke University, 2018. 17, 22

[49] H. Sukhwani, M. Mart, X. Chang, K. S. Trivedi, and A. Rindos, "Performance Modeling of Blockchain Consensus Process (Hyperledger Fabric)," in *IEEE Symp. Reliab. Distrib.*, 2017, pp. 253–255, ISBN: 9781538616796. DOI: 10.1109/SRDS.2017.36. 16

[50] *What is a bundle?* [Online]. Available: https://docs.iota.org/docs/getting-started/0.1/introduction/what-is-a-bundle. 26

[51] *What is a transaction?* [Online]. Available: https://docs.iota.org/docs/getting-started/0.1/introduction/what-is-a-transaction. 25

[52] *What is IOTA?* [Online]. Available: https://www.iota.org/get-started/what-is-iota. 22

[53] R. Yasaweerasinghelage, M. Staples, and I. Weber, "Predicting Latency of Blockchain-Based Systems Using Architectural Modelling and Simulation," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, pp. 253–256, 2017, ISSN: 0012-9658. DOI: 10.1109/ICSA.2017.22. arXiv: 1802.07817. 13

[54] K. Yeow, A. Gani, R. W. Ahmad, J. J. Rodrigues, and K. Ko, "Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2017, ISSN: 21693536. DOI: 10.1109/ACCESS.2017.2779263. arXiv: 1702.05812. 2, 11, 12

[55] Y. Yuan and H. Zhiwei, *DAG Technology Analysis and Measurement*, 2018. [Online]. Available: https://www.reddit.com/r/Iota/comments/8ylryh/dag%7B%5C_%7Dtechnology%7B%5C_%7Danalysis%7B%5C_%7Dand%7B%5C_%7Dmeasurement/. 7

[56] M. Zander, T. Waite, and D. Harz, "DAGsim : Simulation of DAG-based distributed ledger protocols," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 118–121, 2018, ISSN: 01635999. DOI: 10.1145/3308897.3308951. v, 15, 48, 53, 77

[57] Z. Zhang, V. Vasavada, R. King, and L. Zhang, "Proof of Authentication for Private Distributed Ledger," in *NDSS Work. Decentralized IoT Syst. Secur.*, 2019. 21

[58] Z. Zhang, V. Vasavada, X. Ma, and L. Zhang, "DLedger: An IoT-Friendly Private Distributed Ledger System Based on DAG," *arvix*, no. February, 2019. arXiv: 1902.09031. [Online]. Available: http://arxiv.org/abs/1902.09031. 21

Appendix A

Transactions Data Segments Examples

{**hash:** 'NBAEQAUNYTFP9KPLISGCBDEACLAJZXL9NIAGI9BSQBOJ9IZUNB
HSNJH9YWTNZ9BSXMCWCO9XCNBLXY999',
signatureMessageFragment:
'RBTC9D9DCDEAFCCDFD9DSCFA99999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999

99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999999999999999999
9999999',

address: 'FYBWJLPBZBHVRZIUNXKFCYRPAVHYHHWMH9RETQNLPYLOEE
9RIYTKOYVOACWJPNEDLPBTZUOUQEFUPKXKY',

value: 2,

obsoleteTag: 'ZG99999999999999999999999',

timestamp: 1545414204,

currentIndex: 0,

lastIndex: 3,

bundle: 'VHHKWTHEWWJBOSG9CRWINTFJBSGEIQSHIALO9BYTGGPNO99
DXEZ9LSHTAXBELZYKWEZXXKOZYZETYGWVX',

trunkTransaction: '9KBKIBOTIRRJLC9VVHZXTZ9WBDWISIVVKNBBNLZDR
FTNRVGYWM9WLZCOMHQEVLXMIHYRWOFVXFXOYD999',

branchTransaction: 'UCJXFHVZRFJU9FKPBKWEKTJORROVWRIKHPVIWZT

STBJXKQX9PDMCSNSWZJTQEBAOBVAOVXMEAJID9R999',

tag: 'ZG99999999999999999999999',

attachmentTimestamp: 1545414240348,

attachmentTimestampLowerBound: 0,

attachmentTimestampUpperBound: 3812798742493,

nonce: 'LUOMCU9WTCUZBZTOFHNDJURSESW'},

{hash: '9KBKIBOTIRRJLC9VVHZXTZ9WBDWISIVVKNBBNLZDRFTNRV
GYWM9WLZCOMHQEVLXMIHYRWOFVXFXOYD999',

signatureMessageFragment:

'VHZOLWSSLGCTUZEIC9HRERFHKQEBI9GGHGSNFXEBLQFCSAVDAQFZ
M9KBF9TELOUEMR9TYUTDTIAKG9JMDBOHUIMFYYUJSTLRDMUYLGG
9QLBAHFHBVZRZ9OGX9OCVOTHSRVAJYCVDWEKFIXWJXVSVOTGNMI
CGMUCBEZPEUIYTVBENYGTIUMJGVQCXLBUETBMCZVDPAAJ9BVOSH
99UFHYUOBFFRDYJCJYAMQTRMSMGAYYPUWZSJJCJQFVFVSBROFRJQ
Z9XPL9VG9IIUVVCVJWRLBSKJTRSBGFQXRACNQVDLOZNYAQZPTOSNF
YVFXJGGG9J9GWWHLARIQZMIDAOTT9PKAGH9SSSBNTQDMKEI9BYINE
QFYJAXMCQMSESPLOEEQLAXCWNTP9HAE9ETVJYDRNO9NUFCAPNHII
TDEXAABJYFINLLDDKJUMAMMX9TTINHHITJEYITLUDLQRJCOGDGXR
B9YSYLCKDJZONUODKHWONMBIPOXZVAHLAWDASHR9PXVUFQXQLESY
OAHBXKMTPJIMUMSIXBZARVKYSTTAMZSZVGJELDP9UWH9NPFDVUIPW
JLTVQEOYCVZMPHSFMRZJGHACCLLOSBDQWCZEZY9DUBTRMJMSHXEZUK
JHAOFKZUG9KSJGWGVKOLADDRMG9OVUONKTYZTKKTIUDTXUQNZTXQU9
FECUJUXRWPTTUGNIPMNERXFGDFIOFXAU9LWMWTNINFNTJUUYWQCNID
USOSYONTCJAPQP9OTDAEMVJAPKMKGIGS9CANX99JUINDYLFUFLNOUC
AMAKE9V9OFLMYEB9JYCOQ9GGJNSCNZTHBZEVCDUNPXLSMPTPUTRSEX
QGQXVT9AAWUCMHBJUMHRCSTPX9LVWCXURPXBXD9XC9BRLRLQDUC9HK
PGGZXTWTWUQVPI9CMGDPXZMPMSMQUAHVMNALXRFTGKLUUEMY9ILHZD

MY9WXLQWETRUFGAIYLFFSCVHEDWSRGHGQLGOLQBHKRCR9AVCEGLUPY
IKOLMYLJQMSCTCTH9MRLHOAMRSKABKDDNTVJTLBLE9EKATKF9M9OXQ
LDBHDZHCHXYNMVHAOUEPRJTDZMAUSAGAXMVKQOLJ9QZZVTHKPIUFFA
ND9IBLCUZZXC9UXNELVTDCEWXAFMMVOXGP9F9WVKDGCEIFVBNJQSVG
XDJ9GYWYASKVCIQPYQIUQLIDFNKQESHDJRTIYOIXHQJELVFAWQKYCM
WLNSXHPQKWATMCVFTRIABXGDJTYXU9PHFZZAVVZLTOKKL9HHXLQHGF
UTSURCBWQKWHNYMIQESTYBZUGSDKEO9JTEIUPDK9DHSCEOVSYGFBAT
PTIPVZOTZRXYJRVMJGDLBUYTIYEGZKOGPPNZMIJDELAIDIJZWNW9RD
XIYYICABIFOIZROXUJKGNAEJZEAQBCYVKGRVTFX9PFFZMHCHKM9P9O
IAINHQZYSPLOJMMSXGVMN9OSKYWGGNVNM9TJQMNWTOVDZGMQBVRVKD
AAQAWYX9ATVEDZLVWSVSIALPHVKUCUED9EKBDOAXQHIL9YUTX9DI9U
SJRXDSFERDLFM9KV9XFTWFMXHWSIHJSLGVRKOSMMOQIOIWWXJLMLUK
WOVKMPQMJQE9NAIACMZDGZZEHDQZFTQVGAHZEJGEG9GUVVJOTIYXANT
VYLGLV9OVPI9FSVQRHPYFSPJT9QUHZWXWSSVROKHIGD9BUQZLBUGPW
TYRWVHSQUBJTWGNKSVEOUCRLIC99RPNSIEBVZPYBCR9FGZWPJUIUKH
TD9EPVRHTFTKAAESRCRWDZKWWJPM99KRO9PMZXFPKPJBTMYJNEKVIK
XVKZP9WOUFGGXVICNXMPEKKPLSBVHKJGKBXWIPJIPRURCWOAYSQJYU
HYLIZHR9TDVHBAKSDNTUHOZLJSWSOTRGCANMFTYXJKTPKNJQEMURQB
FUFEDPBZFNSCWBXNCDXWXQDFFTPMZDMWRFGNRDSHLKZIKHZLQMHQRV
AAEXHN9ROJ9DALMKPAO9PWAYXUMMMUJDYOCQUNBDFWPXTTYZLNBXEM
SJSVXAJZLWZLMTOCAGFQLZYNOLBDBRBDZLMZIRNPDQCLEUNC9UHTKY
MAQCSCKAEUEWIGECSHXJXGQMJVJACEAA9DNNPEBAIFCCZZALIUAHHG
SUYIYFRBGTILXEZHDTMKRZ9SWLGTRRSUAJLPZXVEYDPEOLVZ',
address: 'VNW9PEERAGDECXTIIPHNPBSCUEEIWGPSQJDBPCYARFQJWXYS
LCRZXYHFRUYCDBKKBPUWAGUKHGADECNCY',
value: -2779530254277755,
obsoleteTag: '99999999999999999999999999',
timestamp: 1545414228,

currentIndex: 1,

lastIndex: 3,

bundle: 'VHHKWTHEWWJBOSG9CRWINTFJBSGEIQSHIALO9BYTGGPNO99D
XEZ9LSHTAXBELZYKWEZXXKOZYZETYGWVX',

trunkTransaction: 'PGTNOTZQFVODZGKRGIRONH9GROAGJPCGGSY9KG
9ZLPRMEQZ9GCY9N9TN9SCULJOWNVXCCFQBRMVHXU999',

branchTransaction: 'UCJXFHVZRFJU9FKPBKWEKTJORROVWRIKHPVIW
ZTSTBJXKQX9PDMCSNSWZJTQEBAOBVAOVXMEAJID9R999',

tag: '999999999999999999999999999',

attachmentTimestamp: 1545414240328,

attachmentTimestampLowerBound: 0,

attachmentTimestampUpperBound: 3812798742493,

nonce: 'WUXQRSOE9UREZJZNODXGQPGVUSH'},

hash: 'PGTNOTZQFVODZGKRGIRONH9GROAGJPCGGSY9KG9ZLPRMEQ
Z9GCY9N9TN9SCULJOWNVXCCFQBRMVHXU999',

signatureMessageFragment:

'GMKUKVLDLEFBH9TVYXWYGTBREOUBOLYEPH9IVZKXELOIHMLVRBF
CENJCBMJAEMFBBVGFCOHYPIZYOIJBCGPGRHSHTVNOQGTSWKRCU
OCQVYQXUPXFAJZLICWKORPEJHOVAW9CYW9PJHBSIIPQBGNTIPG
KIAZTKMSYVDWBZGCUFEGTYUOVVEILAPWEJEVVSNZTOEYQJ9EZR
GPQAE9LETBXLXVXXYVJQWRTMMEHMOERCASORRNNPMXSSPJHWS
ULRWONPM9VNBRRFNRYTXBTYYKUMVZMACKVDH9PBJK9XXDDSN
HNP9DWLGKPEIHRHWXPSFOLTXFCGRNBIAJI9PKIDZAEMILFKCG
WQMZVOCEOBIPYIWFEJFTWCYUGMKMLTFKFQEXTCSGBMPQQFPLO
CLWBDHE9AAXBMSJQR9YHYFBWSDOWCNKOICMHB9ZYXOJKELVEM
KEH9WLOTHWHXKCUVNAJJYYOX9TCF9ESFZGZURNKVWDQYJCVVC
9YZYWNVXRREXSVURCPAEVGUGDXS9PCPPPNLXBMJLBMHSGAJLN

FIVJVZAIVROEGUNNUCHCUDIXDEYQBO9VISSFDERIGHSFJAUCE
FGYAKVI9TGXUXLVXEOVEKSCEOJFVPAKARISUYSECLHVSDPRHW
EDAMDSDX9YOKD9XFXKTC9ILTQOQLOZSUUBENGDJCPTNE9ZMYE
L9AHEKIXFWTHTXHUNPQGUWNQIADHZOYZFPKKNDBWXYMNXOO9Q
UTAULKXQGCFKWJWAHTBKLDZUEWFFFHJQDV9BHOSMZUYYPTAKZ
JZGGFADGKKXNSGATMHGNJXONUVMMBJ9NZKSAZQZERRLDYOPHB
LAQQPNFINJHNMZAYU9WFMZYNDKRCXIMTAXOIEXDYGDJZNXFLC
RUPEXLYHKBTPNJRQWILEVGJUMWIPXUHVSMVLJLZZXHKHPSEDL
NTMDY9UGPBDDYCOEYGPRPGJNOMLKPDHX9SGNBTOWBMFPCKJNT
GYTTFLYNJFCIXJKXQHIIBULLLNOSOGWJCISFMQMZKNGFRHOVG
FPVXAWG9TUBIPGPPED9WUZUAQRWCERHWAJPSVPKJBERPFXQCH
GKJRQECIYRMPFGJJQWNMSZBJCQWUENQSTQGVZKFIEYZNPINGH
WNALDRJSKUUUAUOBWJNTU9AO9F9SSIUYPRTFGIFVROQVUJAEN
MFTWPFXAWBOWGWCLHBRFLJBJ9BZPNKTEYY9NMNDCSLQDDSQNO
AEFSYDSJOBTNRQLXHP9I9SDIMAAAPFIPJZLRNHTXMTPGKYVAH
KRERZGJJFVBEKOBCOYT9AJMFXXFVNKZXCAOLJQQXRND99LXTX
XGLKDHGXRHZZJBUQAOKVO9EN9WUHDFEILAHULPQUIXRQSRALN
BFHYWNKWWNYEMSFD9XDLGKFEQLZGGQIYUVRTASME9YNPUJKLD
EEDEFONHXIEJTMSYEWQTWKCDNEKTFIZPWVKTWKUQFPDLPEAFX
OXXFQCFYBWGYTICKCBYNZWHTNXGMAQDN9AWMCFPIXJGRXCI9J
GDWTUM9FDUXBKEQDEOFPUUUBJIMFQHDLGJPZEAZZTEICFABZK
XKKUVFTCCWWSPDMHSBAFIQCNRNPFWJAIWBJBMGLACT99GVBCV
U9XWFOMFXM9SLJJEMPOVYYTWSOTZRF9XAEURODGRHWRJ9LXA
ZDFJYBCEG9AVOT9ZQIPVOLJGXZFGKFJCPZIUFZWBTSIVIWUE
IVFSIMS9SBCJ9BYHSTOSHXTJCVCMBKQMIPUVWMADERXNE9U
PAEFXPGSRFTY9FOHWXHNDKCCOIMXCSRSEPVTGEORJJASCEE
IVHZVOFKDZYWUQHXMHKOTOTVPURMQ9RCPDGKQDIPR9XUHWU
POFQCWHDHNYZCZVDRNTPYEHLDJVTXQHWHJUAHHAVWPROEME

RYRRZQKFBZUWEYOSOCHPSQBXBGDZDGSKSBGEXCXOOHLJEKV
BFSGWSSLYZLJWGXD9REFGOESJMONNNIBRRJYKTRXFAUEWKG
LMOC9NRJKRVJVRVROJPJBVFEQDWJMHCLIPXNNVDVSNNEXZT
GSHKGJFVZNGBGCGRKWPSBMBZHNJDSZJGNKA9ABIDNXGBBFV
GSLGZNFPDSC9AIOCORYGRBLODSYTNSWVWQWDXQQLXHKMGNL
CBTSHNTFYJJQDSWVOCPVHUUPCBVVECTFOTAOH9KMPEWFYDB',
address: 'VNW9PEERAGDECXTIIPHNPBSCUEEIWGPSQJDBPCYARFQJW
XYSLCRZXYHFRUYCDBKKBPUWAGUKHGADECNCY',
value: 0,
obsoleteTag: '999999999999999999999999999',
timestamp: 1545414228,
currentIndex: 2,
lastIndex: 3,
bundle: 'VHHKWTHEWWJBOSG9CRWINTFJBSGEIQSHIALO9BYTGGPNO99
DXEZ9LSHTAXBELZYKWEZXXKOZYZETYGWVX',
trunkTransaction: 'IDEUQSJK9NZCVI9MSSGIFGRPCHDKMVMYCONIWWCDO
PFEH9ANHBRLKPQKEANJRKZSLJEDQUYYDC9YNN999',
branchTransaction: 'UCJXFHVZRFJU9FKPBKWEKTJORROVWRIKHPVIWZTS
TBJXKQX9PDMCSNSWZJTQEBAOBVAOVXMEAJID9R999',
tag: '999999999999999999999999999',
attachmentTimestamp: 1545414240233,
attachmentTimestampLowerBound: 0,
attachmentTimestampUpperBound: 3812798742493,
nonce: 'QLXPXKFHQI9UYMRC9CGLJBJTBZR',

{**hash:** 'IDEUQSJK9NZCVI9MSSGIFGRPCHDKMVMYCONIWWCDOPFEH9
ANHBRLKPQKEANJRKZSLJEDQUYYDC9YNN999',
signatureMessageFragment:

’999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999999999

9999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999’,
address: ‘XWTJZJXANWCGAWB9QMGZSTHPFZTZXHQJKLYVTGFBGVSXW
XENNBWNCZMYRVNKZIPZDGAWN9KTRU9ACXTMW’,
value: 2779530254277753,
obsoleteTag: ‘999999999999999999999999’,
timestamp: 1545414237,
currentIndex: 3,
lastIndex: 3,
bundle: ‘VHHKWTHEWWJBOSG9CRWINTFJBSGEIQSHIALO9BYTGGPNO99
DXEZ9LSHTAXBELZYKWEZXXKOZYZETYGWVX’,
trunkTransaction: ‘NBAEQAUNYTFP9KPLISGCBDEACLAJZXL9NIAGI9BSQB
OJ9IZUNBHSNJH9YWTNZ9BSXMCWCO9XCNBLXY999’,
branchTransaction: ‘UCJXFHVZRFJU9FKPBKWEKTJORROVWRIKHPVIWZT
STBJXKQX9PDMCSNSWZJTQEBAOBVAOVXMEAJID9R999’,
tag: ‘999999999999999999999999’,
attachmentTimestamp: 1545414240194,
attachmentTimestampLowerBound: 0,
attachmentTimestampUpperBound: 3812798742493,
nonce: ‘NIHCJNERVUGSQQXZWEPXQDHUGDS’}

Appendix B

CurveExpert Fitting Results

$\lambda = \mathbf{1}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 7.597946914609123E+00

b = 7.943828433039338E+00

c = 4.245743460176508E+00

Standard Error : 8.637908580008021E-01

Correlation Coefficient : 9.339533235969112E-01

Run time : 0.0023 seconds

$\lambda = \mathbf{2}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 1.656140886252584E+01

b = 8.359843020670617E+00

c = 3.516505769978269E+00

Standard Error : 9.088597026680623E-01

Correlation Coefficient : 9.883460773974604E-01

Run time : 0.0044 seconds

$\lambda = \mathbf{3}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 2.243901434563289E+01

b = 8.975977898915819E+00

c = 3.741940152771595E+00

Standard Error : 1.274139081478851E+00

Correlation Coefficient : 9.871259635278757E-01

$\lambda = \mathbf{4}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 3.208734901892893E+01

b = 9.171769644309206E+00

c = 3.655720068037237E+00

Standard Error : 2.271282575567261E+00

Correlation Coefficient : 9.822652693089272E-01

Run time : 0.0104 seconds

$\lambda = \mathbf{5}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 4.023210001443740E+01

b = 9.726190073776058E+00

c = 3.296228210549058E+00

Standard Error : 1.911441733256013E+00

Correlation Coefficient : 9.921784070452034E-01

Run time : 0.0049 seconds

$\lambda = \mathbf{6}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 4.937829985016278E+01

b = 9.389546386444806E+00

c = 3.298030861201210E+00

Standard Error : 1.475601805220076E+00

Correlation Coefficient : 9.967759824414489E-01

Run time : 0.0045 seconds

$\lambda = \mathbf{7}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 5.180392255621111E+01

b = 1.016873068538634E+01

c = 3.859224020937584E+00

Standard Error : 3.469219121565799E+00

Correlation Coefficient : 9.838821707162396E-01

Run time : 0.0025 seconds

$\lambda = \mathbf{8}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 6.669069181229767E+01

b = 9.745413916316956E+00

c = 3.295814455829923E+00

Standard Error : 2.241793292643348E+00

Correlation Coefficient : 9.961342385661439E-01

Run time : 0.0031 seconds

$\lambda = \mathbf{9}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 7.102122246097127E+01

b = 1.029821570822472E+01

c = 3.536297704461834E+00

Standard Error : 3.733086867023691E+00

Correlation Coefficient : 9.903385254893218E-01

Run time : 0.0057 seconds

$\lambda = \mathbf{10}$

Autoscaling graph Top Results...

Distributing the calculation over 4 cores...

Final Result [Miscellaneous/Gaussian Model]:

Equation : a*exp(-(x-b)^2/(2*c^2))

a = 8.100872758386177E+01

b = 9.721720331324763E+00

c = 3.282548517071222E+00

Standard Error : 4.146295601169974E+00

Correlation Coefficient : 9.909119046119099E-01

Run time : 0.0034 seconds

Appendix C

Compass and IRI Configurations

ubuntu@tangle-iri-node20: /compass/docs/private_tangle$ cat config.json

{ “seed”: “VXFJVCC9SPVZSEVRDSKKHRTDNOMJ9KXRKVKTOG

9UHIGR9EWOXVNDFEDCAQAUJCLRSRWOBKBKB9POFVQUW”,

“powMode”: “CURLP81”,

“sigMode”: “CURLP27”,

“security”: 1,

“depth”: 16,

“milestoneStart”: 0,

“mwm”: 9,

“tick”: 60000,

“host”: “http://localhost:14265”

}

ubuntu@tangle-iri-node20: /compass/docs/private_tangle$ cat 02_run_iri.sh

#!/bin/bash

scriptdir=$(dirname “$(readlink -f ‘$0’)”)

.$scriptdir/lib.sh

```
    load_config

    COO_ADDRESS=$(cat $scriptdir/data/layers/layer.0.csv)

    docker pull iotaledger/iri:latest
docker run -t –net host –rm -v $scriptdir/db:/iri/data -v $scriptdir/
snapshot.txt:/snapshot.txt -v $scriptdir/iri.ini:/iri.ini -p 14265 iotaledger/iri:latest

    –testnet \

    –remote \

    –remote-limit-api removeNeighbors \

    –testnet-coordinator $COO_ADDRESS \

    –mwm $mwm \

    –milestone-start $milestoneStart \

    –milestone-keys $depth \

    –snapshot /snapshot.txt \

    –config /iri.ini \

    –max-depth 1000 $@
```

ubuntu@tangle-iri-node20: /compass/docs/private_tangle$ cat iri.ini

[*IRI*]

API_HOST = 10.12.7.43

TCP_RECEIVER_PORT = 15600

UDP_RECEIVER_PORT = 14600

NEIGHBORS = udp://10.12.XX.XX:14600 ... udp://10.12.XX.XX:14600

ZMQ_ENABLED = true

ZMQ_PORT = 5555

REMOTE_LIMIT_API = "removeNeighbors"