ANALYSIS OF CONSENSUS PROTOCOLS ON LATENCY IN BLOCKCHAIN

Author

Rabia Basharat

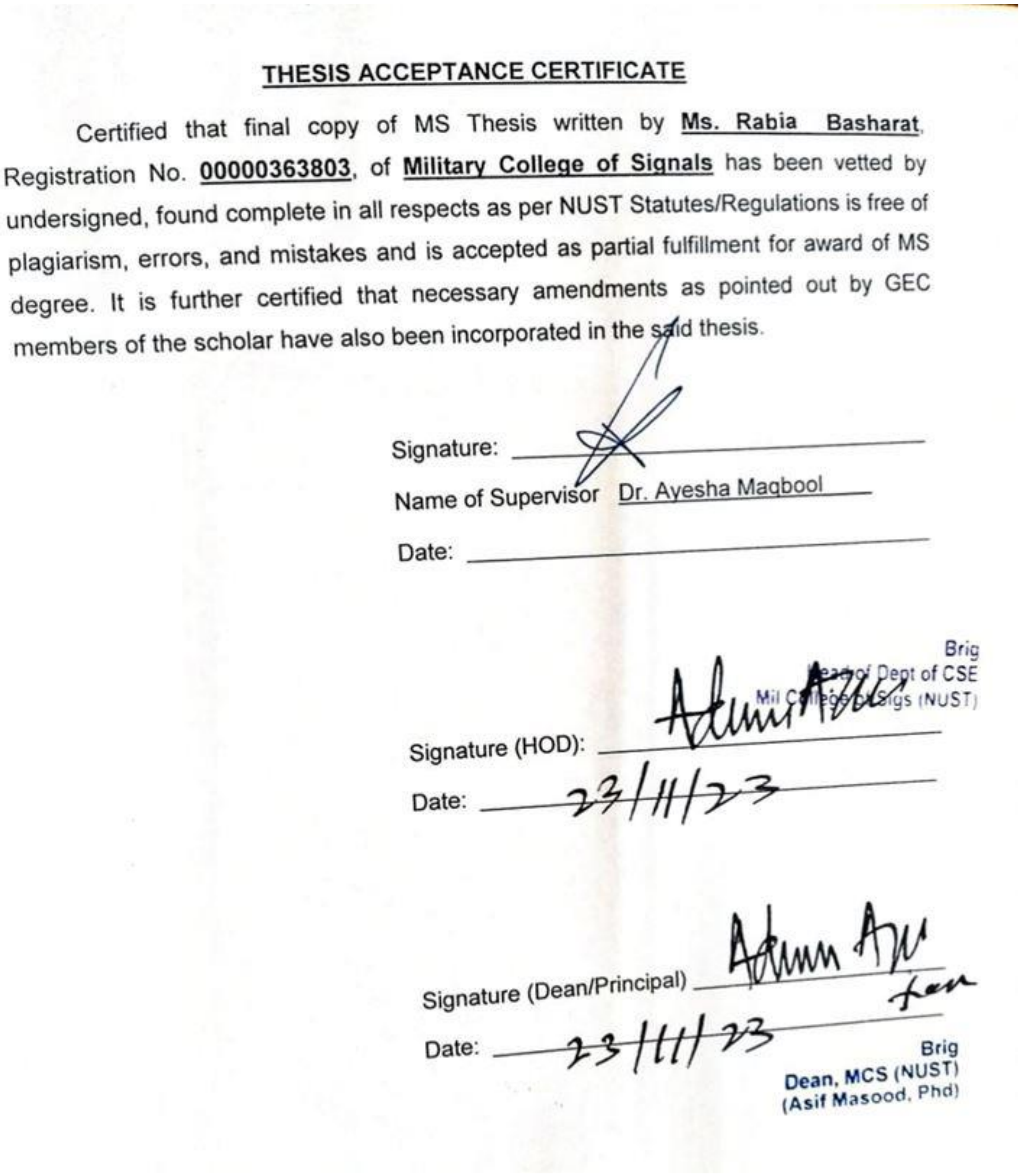
00000363803

Supervisor

Assoc Prof Dr.Ayesha Maqbool

A thesis submitted to the faculty of Department of Computer Software Engineering, Military College of Signals, National University Of Sciences and Technology (NUST), Rawalpindi, in partial Fulfillment of the requirements for a degree of MS in Computer Science.

November, 2023



THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Ms. Rabia Basharat**, Registration No. **00000363803**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ______________

Name of Supervisor Dr. Ayesha Maqbool

Date: ______________

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

Signature (HOD): ______________

Date: 23/11/23

Signature (Dean/Principal) ______________ for

Date: 23/11/23

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

DECLARATION

I, *Rabia Basharat* declare that this thesis titled ”Analysis of consensus protocol on latency in blockchain ” has not been already submitted for a degree or some other qualification at NUST or some other institution.

Rabia Basharat
00000363803

DEDICATION

Dedicated to my parents, Gulam Sughra & Basharat Mehmood and my siblings and friends whose constant support and exceptional cooperation led me to this accomplishment.

ABSTRACT

Blockchain, a cutting-edge technology, stands as an immutable ledger, housing various data types and providing a platform for managing and tracking asset ownership. Woven together by principles such as cryptography, ledgers, immutability, group consensus, and trustlessness, blockchain amalgamates diverse concepts and technologies . This thesis delves deep into the realm of blockchain consensus protocols, conducting an exhaustive analysis of their strengths and weaknesses. A key focus lies on addressing the critical facet of latency, leading to the development of analytical models individually tailored to each protocol. Through the implementation of sequences and map diagrams, the intricate workings of different consensus mechanisms are meticulously unraveled.

A significant milestone in this research journey is the creation of a blockchain-based simulation model, meticulously crafted for the purpose of testing both Proof-of-Stake (PoS) and Proof-of-Work (PoW) consensus protocols. The outcomes of the simulations undergo a rigorous comparison with analytical predictions, resulting in a remarkable alignment that contributes not only to a nuanced understanding of consensus mechanisms but also establishes the reliability of the analytical frameworks employed. This study is not merely an exploration but a pioneering advancement in the landscape of blockchain studies, offering insights that echo across industries and sectors.

ACKNOWLEDGEMENT

On the road to success, there is always we, not me. There is always another who is standing behind you, maybe not coming on the forefront but always at your back. Praying for you and supporting you through it all.

I am thankful to my LordAllah S.W.T to have instilled the strength and courage in me to work. For He is the one who has guided me throughout this work at every step and for every new thought which came to me. Indeed, I could never have done nothing without the will and help of the Almighty.

I would also like to express special thanks to my supervisor Dr. Ayesha Maqbool for their help and support throughout my thesis. I would also like to pay special thanks to them for tremendous support and cooperation. Each time I got stuck in something, they came up with the solution and motivated me when I was lost and confused. Without their help I wouldn't have been able to complete my thesis. I appreciate their patience and guidance throughout the whole thesis.

I would also like to thank my GEC Members Asscoc Prof Dr. Ihteshamulislamand Asst Prof Dr. Yawar Banjash for always being available for help and support throughout my degree in coursework as well as in thesis.

Finally, I would like to express my gratitude to the three people who have been cheering me on and being my firm support since day one; my beloved parents, siblings and my friends , for their patience, love and support through all the tough times and for never giving up on me. It is solely because of them that I was able to dedicate myself to my work and give my best while they handled everything else

TABLE OF CONTENTS

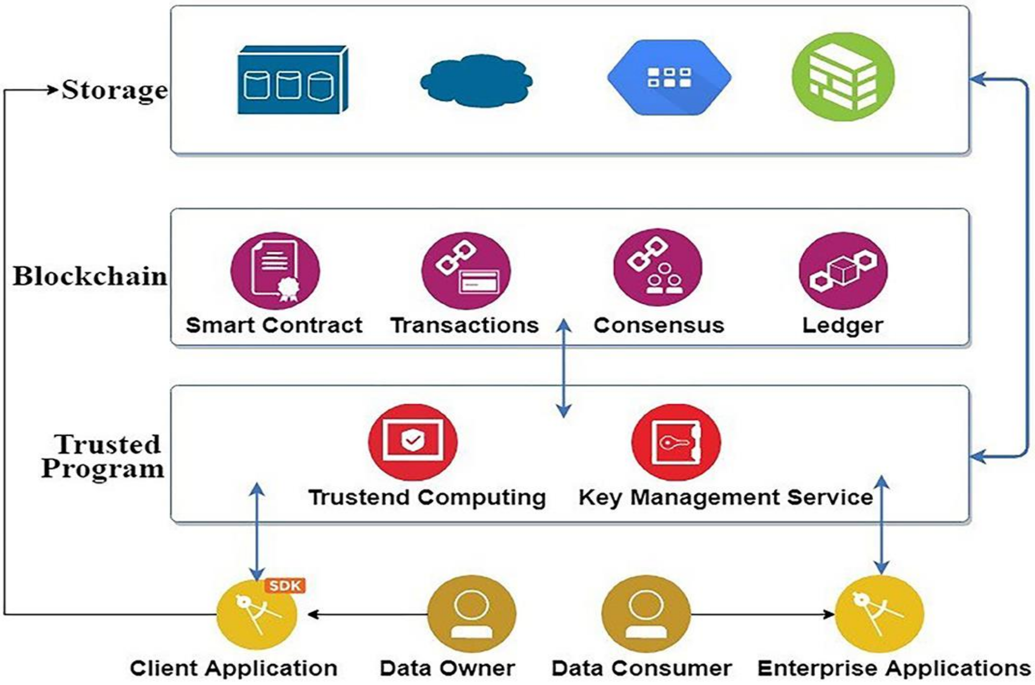
LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Traditional transactions are all dependent on a single trusted person, which creates a number of issues with transaction cost, efficiency, and security. We need to present the idea of blockchain technology in order to address these issues and achieve safe, quicker, and transparent transactions.Satoshi Nakamoto invented the blockchain technology [1]. Blockchain is a new emerging technology for decentralized and sharing of transactional data across a large peer to peer network, where untrusting members can interact with each other directly and verifiably. Blockchain is another innovation with significant implications for the future of how we exchange data and money as a fully organized society**.** One example of a blockchain technology use in the financial sector is Bitcoin. A distributed ledger system is all that the blockchain is. Without the participation of a third party, it will handle transactions between people and organizations**.**



\ Figure1.1 General Architecture of blockchainTechnology

The above figure represent the Architecture of the Blockchain Technology. Here we will discuss the element of blockchain technology which are:

1. **Ledger:** Blockchain is a distributed ledger system, which implies that everyone using the network has a copy of the same record. In the Blockchain, neither a centralized authority nor a reliable third party exists.
2. **Consensus protocol:** All the transaction should be verified by all the parties in the network The process of producing a block and adding to the network's ledger is known as mining.
3. **Security:** Blockchain makes use of public key cryptography and digital signatures to confirm the authenticity of network transactions**.**
4. **Cryptocurrency (or cryptocurrency):** It is created as a digital asset that functions as a means of exchange for delivering safe transactions using encryption.
5. **Privacy:** The blockchain may be used to store any kind of data. If sensitive data is processed, such as health information or citizen services, the privacy regulations

apply.

6. **Smart Contract:** These agreements are actions having the capacity for self-execution and self-enforcement. These contracts rely on data from outside sources, thus in order to prevent data tampering, a cryptographic proof must be provided.

1.1 Blockchain Technologies

Blockchain is a technique for storing data that makes it difficult or impossible for the system to be altered, hacked, or otherwise abused. A blockchain is a type of distributed ledger that distributes and copies transactions throughout the network of computers involved.Blockchain technology is an organisational framework that maintains public transactional information, also known as the "block," in several databases, also known as the "chain," in a network connected by peer-to-peer nodes. This type of storage is frequently referred to as a "digital ledger." Every transaction in this ledger is validated and protected against fraud by the owner's digital signature, which also serves to authenticate the transaction. Consequently, the data in the digital ledger is quite safe.

A block is a component of the blockchain, which records all transactions and, after they are finished, adds them to a running database. Blocks in Blockchain are connected one after the other like a linked list. Each block contains the preceding block's hash.

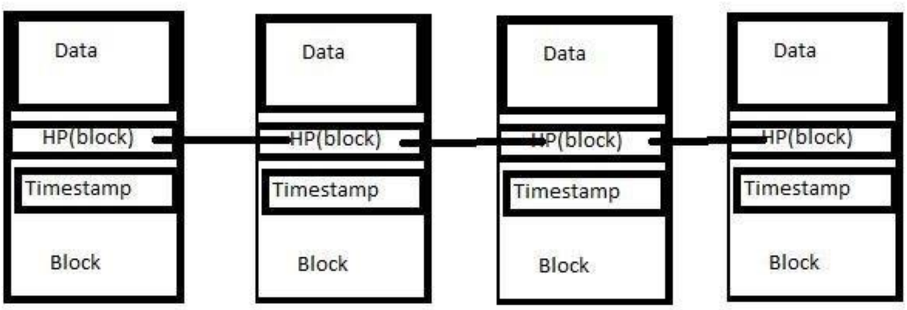


Figure 1.2 Blockchain as A Linked List of Blocks Connected by Hash Pointers

A network of nodes constructed like a peer-to-peer network makes up a blockchain. Users can communicate with one another on the blockchain by using public and private keys. They address the private key on the network with the public key and use it to sign their own transactions. In the network, it offers authentication, integrity, and non-repudiation.

Before sending further, each node in the blockchain verifies that the incoming transaction is legitimate. Transactions that aren't legitimate are ignored. There should be a set of rules for every database transaction in any Blockchain network. Each blockchain client has these rules pre-programmed, which are used to determine if an incoming transaction is legitimate or not.

1.1.1 Types of Blockchain

Blockchains are classified into three types.

1. Public Blockchain,

Public blockchains served as the foundation for Bit coin and other crypto currencies and helped spread awareness of distributed ledger technology (DLT). Public blockchains also aid in removing some difficulties and problems, including as centralization and security weaknesses. Instead than being kept in one place, data is spread throughout a peer-to-peer network using DLT. For authenticating information, a consensus procedure is used; proof of stake (PoS) and proof of work (PoW) are two popular consensus techniques.

2. Private Blockchain

Companies can use private blockchains to customize their accessibility and authorization preferences, network parameters, and other crucial security options. Private blockchains operate on closed networks, and tend to work well for private businesses and organizations, In a private blockchain, only the blockchain's owner has the power to change the data, while the other nodes only have restricted access. The Private Blockchain employs the PBFT consensus algorithm.

3. Permission Blockchain

Permission blockchain networks, often referred to as hybrid blockchains, are private blockchains that grant authorized users unique access. These kinds of blockchains are frequently set up by businesses in order to achieve the best of both worlds. They also provide greater structure when determining who may join in the network and in what transactions.

1.1.2 Technologies involved in Blockchain

Public Key Cryptography is one of the technologies used in Blockchain technology and it is used for encryption and decryption of sensitive data and message authenticity.

Encryption and Decryption

Asymmetric encryption makes use of public and private key pairs. The private key is used to decode data, whereas the public key is used to encrypt data. Let's say that Alice and Bob are in communication, and Alice sends a message to Bob that has been encrypted using Bob's public key. Bob used the message's own private key to decode it. The attacker needs both private keys in order to learn about Alice and Bob's conversation.

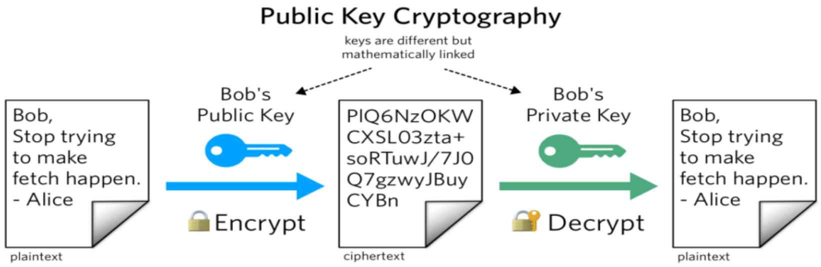

Public Key Cryptography
keys are different but mathematically linked
Bob's Public Key
Bob's Private Key
Bob, Stop trying to make fetch happen. - Alice
plaintext
Encrypt
PIQ6NzOKW CXSL03zta+ soRTuwJ/7J0 Q7gzwyJBuy CYBn
ciphertext
Decrypt
Bob, Stop trying to make fetch happen. - Alice
plaintext

Figure 1.3 Public key Cryptography

Digital signatures

Additionally, PKC can be used for authentication. The recipient can verify the digital signature using the sender's public key once it has been created using the sender's private key.

Hash functions

It is nothing more than a mathematical function, known as a hash, that converts data of any arbitrary size to a certain set length. Hash functions are one-way operations that never get the original data; they only ever receive the hash value. Additionally, since the same data always yields the same hash result, it is deterministic.

Homomorphic encryption

With homomorphic encryption, users may execute binary operations on encrypted data without ever having to decode it.Without having access to the raw data, this type of encryption enables the encryption of data before sending it to cloud services or processing environments for example

Arithmetic operations can be carried on encrypted values. Plaintext1 with Encryption changed to Ciphertext1 Plaintext2 with Encryption changed to Ciphertext2

Ciphertext1 * Ciphertext2 = Ciphertext3

Ciphertext3 with decryption converted to Plaintext3

Plaintext1 * Plaintext2 = Plaintext3

1.1.3 **Consensus Mechanisms in Blockchain**

The consensus technique gives the transactions a clear sequence and validates the transaction block, in blockchain application two problems need to be solved 1) Double Spending Problem 2) Byzantine Generals Problem.

Double spending problem

The possibility of using a cryptocurrency more than once is known as double-spending. A blockchain's transaction data may be changed under certain circumstances. If the prerequisites are met, updated blocks may enter the network, and the individual who made the change may recover any spent bitcoin

Byzantine Generals Problem.

Data is sent between nodes in a distributed system using peer-to-peer communication, but there is a potential that some of the nodes might be attacked, changing the nature of the connection. As a result, we need to identify the normal nodes. Strong consensus algorithms must be designed in order to tackle these consensus .in next session we will discuss all the consensus algorithm.

1.1.4 Security issues involved in Blockchains System

Blockchain systems, despite their inherent security features, are not entirely immune to vulnerabilities and security issues Blockchain systems mainly to look at security in the following aspects

1) Ledger Level Security

Only authorised members are permitted to use the blockchain. The members' transaction must be signed and legitimate users start transactions in the network.

2) Network Level Security

From a network perspective, communication between parts of various nodes must be safe. It must be protected against a variety of network-wide internal and external attack vectors. The ledger need to be able to survive denial-of-service assaults

3) Transaction-Level Security

PKI ideas must be used to encrypt transactions to prevent data breach with unauthorised parties. Identity and authorisation of transaction generation must be protected,only transactions using a certain name 'X' may be carried out.The transaction information cannot be changed or altered, and the multisignature capability is accessible for delicate transactions in the blockchain.

4) Associated Surround System Security

Shadow databases and other related system components should only be accessible by authorised users. Implement authentication and authorisation techniques to do this. It also entails document sharing to guard against malware, worms, and viruses.

1.1.5 Privacy issues in blockchain systems

The cryptocurrency Bitcoin is a prominent blockchain. Because it includes a permission-less blockchain ledger, anybody can see and confirm every transaction.It appears to violate each user's right to privacy.The two types of privacy mechanisms in blockchain systems are transactional privacy and unlinkability**.**

1) Transactional privacy

The only people who should have access to the transaction information are the parties to the transaction, any regulators, and auditors. Participating nodes have a method for validating transactions when there are money available even when the transaction is completely encrypted.

2) Unlinkability

Random entities unable to get information about transactions with others. In order to obtain information on the parties engaged in the transactions, it would be feasible to mine data from a number of transactions. Unlinkability aims to prevent such inferences from being drawn**.**

1.2 ProblemStatement

As technology advances day by day, people have tried to create innovations in all areas of science so far,they have made revolution in the field of NFT domains. Blockchain technology has been considered a breakthrough for the many system research domain. Blockchain is the method of recording information which is impossible to change hack or manipulate, it contains distributed ledger that is used to store data in a secure, transparent, and tamper-proof way. Blockchain described as decentralized technology and peer to peer distributed system It uses distributed computing and cryptography to securely host applications, store data, and easily transfer valuable digital instruments that represent real world money One of the main characteristics of block chain is its decentralized nature. That is, it is controlled by a network of users rather than a single entity. This makes it resistant to tampering and hacking, as any change to the ledger requires consensus of the majority of the network. Blockchain technology is a blooming technology with so many potentials and benefits. But there are a lot of concerns that are detrimental to its adoption. These include scalability, privacy leakage, selfish mining, transaction malleability, high electricity cost, absence of standardization, limited interoperability among blockchain networks, and of course latency. Latency has to do with the processing time and transaction the time it takes each transaction on the blockchain to be executed. This is a very important parameter because it suggests how fast a blockchain network. There are different consensus protocol susedin blockchains, and each protocol has different latencies. People prefer low-latency, high- throughput protocols, so we are going to do critical analysis of consensus protocol regarding latency and this is an un researched approach in blockchain,and we contribute by filling the gap.

1.3 Reason/ Justification for the Selection of the Topic

As blockchain is growing field and boom of digital currency make it more popular. Since blockchain is a relatively new technology, there is still a lot to learn about how it operates and how it may be made better. Research can aid in our understanding of the blockchain's workings and point us in the direction of methods to improve it. Although blockchain has previously been used in a variety of use cases, there are probably many more that are still undiscovered. We can find new applications for blockchain through research, particularly in sectors like banking, healthcare, and supply chain management. While blockchain has numerous benefits, there are also some restrictions and difficulties. Blockchain, for instance, can be expensive and cumbersome to use in some circumstances. Blockchain can be more useful for a larger range of applications by addressing these restrictions, which can be determined through research.Blockchain technology is intended to be secure, but there is always a chance for flaws and assaults. Research can be used to pinpoint potential security issues and create mitigation plans. There are a variety of blockchain platforms, each with its own set of protocols and standards. By enabling interoperability between several blockchains, research can point to methods to improve communication and data transfer between them. So as a researcher we need to fill the research gap regarding any technology so in blockchain we identified the research gap blockchain regarding the latency of consensus protocol so we are going to do critical analysis of consensus protocol regarding latency and fill the gap

1.4 RelevancetoNationalNeeds

Blockchain technology has the ability to support national needs in a variety of ways, including the following:

Enhancing Government Efficiency: Blockchain can offer a safe and open method for managing data and transactions for governments. It can aid in lowering red tape and the need for middlemen, as well as improving accuracy, auditability, and accountability in governmental operations. Improving

National Security: To improve national security, blockchain can aid in the creation of secure digital identities, the protection of vital infrastructure, and the tracking of supply networks. Moreover, it can help in identifying and averting cyberattacks, fraud, and other security risks.Increasing Economic Development: Blockchain can open doors for efficiency, growth, and innovation across a range of industries. It can speed up and secure cross-border payments, streamline business operations, and lower transaction costs, fostering global trade and economic growth.

Improving Education: Blockchain can make it easier to create safe, verifiable digital credentials like diplomas and certificates, which can help to lower fraud and increase the openness and legitimacy of educational institutions.

As a result, blockchain technology can benefit the nation by providing secure and efficient solutions for a range of industries, enhancing governmental operations' accountability, transparency, and effectiveness, promoting economic growth, enhancing national security, and improving public services like healthcare and education. As a result, we help make it easier to use and, as researchers, we fill in the gaps where technology is lacking.

1.5 Area of Application

There are several uses for blockchain technology, including:

- Cryptocurrencies: The underlying technology that makes cryptocurrencies like Bitcoin, Ethereum, and others possible is called blockchain. Blockchain is used by these digital currencies to offer decentralized and secure transactions.
- Blockchain technology can be used to trace products as they move through the supply chain, bringing accountability and transparency. This can lower fraud, raise product quality, and boost productivity.
- Blockchain technology can be used to generate safe and unchangeable digital IDs. This can be helpful in areas like voting and access control and can assist avoid identity theft and fraud.
- Self-executing contracts known as "smart contracts" can be carried out on a blockchain. They can automate procedures and exchanges, obviating the need for middlemen and boosting productivity.
- Healthcare: Blockchain can be used to safely store and transfer medical data, improving provider communication and collaboration. Also, it can improve patient privacy and assist stop medical fraud.
- Energy management can be made more effective and sustainable by using blockchain to track energy output and use.

These are only a few of the many applications that blockchain technology can be used in. New uses are likely to appear as technology progresses

1.6 Advantages

As part of our examination of the consensus protocol's latency, including how it affects the functionality of blockchains overall A thorough examination of the consensus protocol's latency can have the following benefits:

- Enhanced efficiency: Potential bottlenecks and inefficiencies can be found and fixed by examining the latency of a consensus process. As a result, a protocol may become more effective, processing transactions more rapidly and cheaply.
- Improved scalability: Investigating latency can also reveal scalability problems that might develop as the network expands. Early attention to these problems will enable the protocol to be built to support more transactions and users in the long run, increasing scalability.
- Enhanced security: Investigating delay critically can reveal potential security flaws in the consensus system. These flaws can be fixed.
- Improved user experience: With quicker transaction times and less transaction costs, a more effective and scalable consensus mechanism can improve user experience.
- Competitive advantage: By reducing latency and enhancing the consensus protocol's effectiveness, blockchain networks can outperform rival networks that might not have made similar investments.
- Therefore, a thorough examination of consensus protocol latency can lead to a blockchain network that is more effective, scalable, and secure, which can ultimately be advantageous for users, companies, and the larger blockchain ecosystem

1.7 ThesisOutline

Theoutlineforchapterscanbegivenasfollows:

- Chapter 1: Introduction and objectives of Blockchain.
- Chapter 2:This chapter comprises the literature and background along with abriefdescriptionofexistingtechniquesandapproachesused to calculate the latency of consensus protocol in blockchain

- Chapter 3: Proposed Methodology to calculate the latency of all consensus protocol in Blockchain

- Chapter 4: This chapter contains a model of blockchain where we simulation consensus protocol.

- Chapter 5: This chapter show the result of simulating consensus protocol regarding latency

- Chapter6:Thischapter did the cross validation of result with our drive analytical models

- Chapter7:Thischaptercontains the conclusion and future work of thesis

CHAPTER 2

Literature Review

2.1. Overview

The research explores an effective and efficient technique to calculate the latency of consensus protocol in blockchain through story points. For productive research it is worth analyzing existing work made by different researchers in this regard. For this research to be furthered, background knowledge is provided by the literature research done in this chapter. This chapter highlights several techniques and methods applied to this subject of research; This section presents a brief depiction of existing work in the domain of blockchain technology by doing criticalanalysis of consensus protocol to calculate the latency

2.2. Related Work

When conducting the literature review of the analyzing the latency of consensus protocol in blockchain through story points using different analytical model.anaylsing the latency using frameworks, hardware configuration empirical studies comparative studies and case studies are considered

In 2019 Murat Kuzlu Manisa Pipattanasomporn and saifur Rehman stressed about the analysis of a latency and throughput on the framework of blockcahin which is hyperledger Fabric. By conducting this research they reached out the point that is for latency and scalability calculation, blockchain network depends on hardware configuration. The test result may be different in different testing environments. The finding of this paper may help us to select the suitable hardware configuration as well as blockchain networks that can support as in particular implementation and requirements [1]

In 2019 Luming Wan ,David Eyers and Haibo Zhang They look into how different network configurations may affect blockchain security. To quantify blockchain security in their simulation results, they introduce the concept of global block convergence. The simulation shows that block convergence time increases proportionally as network latency increases, but there is no discernible relationship between block convergence speed and either the size of the network or the difficulty of the mining. It also illustrates that network latency variance is a significant factor[2].

In 2018Abdul Wahab and Waqas Memood they analyse how each consensus protocol operates and become anxious Since distributed ledger technology is a disruptive technology, it has become quite popular. There is a consensus protocol that drives every outstanding distributed ledger implementation. They examined a few well-known consensus procedures in their work. Every consensus mechanism involves certain trade-offs in terms of security, scalability, efficiency, and performance. However, they are all there to support a same goal, which is to stop duplicate spending in a distributed ledger[3]

In 2019 Lei Yang and Xuechao Wang analyses the Low Latency Proof of Work ConsensusIn this work, they introduce Prism++, the first PoW consensus mechanism that can actually achieve Bitcoin-level security, high throughput, and low latency. They develop a new confirmation rule using a revolutionary process that involves explicitly identifying the worst-case attack while co-designing the rule. They demonstrated the importance of this confirmation criterion in obtaining minimal latency in real-world circumstances. They have transformed the 12 Prism++ protocol into a productive software system in the second section of the study. With security on par with Bitcoin, their solution handles over 80,000 transactions per second with a confirmation delay of a few seconds. Our findings support the theoretical analysis of the new confirmation rule and emphasise the significance of enhancing[4].

In 2017 Zibin Zheng1, Shaoan Xie1, Hongning Dai2 In this paper, they emphasized the difficulties and upcoming work on blockchain. The four main qualities of blockchain decentralization, persistency, anonymity, and auditability have demonstrated their potential to revolutionize traditional industries. They provide a thorough introduction of blockchain in this essay. They begin by providing a general review of blockchain technology, including its architecture and salient features. After that, they talk about the typical consensus mechanisms employed by blockchain. We looked at and contrasted these protocols in various ways. In addition, they outlined a number of difficulties and issues that might obstruct blockchain growth and outlined some current solutions. Also suggested are some potential directions for the future. There are more and more blockchain-based applications emerging nowadays, thus they want to perform extensive research on them in the[5].

In 2021M gracy and b.rebbeca stressed the current challenges in blockchain and suggest the solutions Blockchain's stability is ensured by cryptographic technology. Although blockchain is rapidly developing and providing us with a secure and convenient service, it is not without its drawbacks. Bitcoin suffers from low throughput and high latency, both of which are detrimental to the scalability of the blockchain.in this paper we suggest the solution regarding the high latency, our proposed framework helps to do the mining work fast as it motivates with higher rewards and transaction allocation allows miners to select the transaction from the pool to be mined[6].

In 2016bojana koteska, elena karafiloski and anastas mishevalsoaddresses. Latency as an issue regarding the performance of blockchain. Time factor is one of the most critical issues in Blockchain implementations. Having the requests processed on Internet almost immediately, it is an obstacle in regards to the universal technology acceptance.. In order to provide security, the Bitcoin transaction block, the time needed to complete one transaction is about 10 minutes. For a larger transfer amounts, the cost of a double spend attack can last about an hour. VISA transaction completion process takes seconds at most[7].

In 2018Matthias Fitzi Peter Gazi Aggelos Kiayias and Alexander Russell the aim to write this paper to tackle the problem of high latency and low throughput in block chain and suggest suitable solution. As we highlight problem so we suggest solution regarding this problem is the concept of parallel chains the technique by providing two parallel-chains protocol variants, one for the PoS and one for PoW setting, that exhibit optimal throughput under adaptive fail-stop corruptions while they retain their resiliency in the face of Byzantine adversity assuming honest majority of stake or computational power, respectively We also apply our parallel-chains composition method to improve settlement latency; combining parallel composition with a novel transaction weighing mechanism we show that it is possible to scale down the time required for a transaction to settle by any given constant while maintaining the same level of security[8].

In 2022Francesc Wilhelmi, Sergio Barrachina-Munoz and Paolo Dini try to solved the problem of transaction rate in blockchain by introducing the optimal block size in blockchain.In order to develop secure, auditable, decentralized apps, a number of technical issues must be resolved. In this letter, we emphasize the delay that is with blockchain networks that use Proof-of-Work (PoW), where users agree to validate new data before it is appended to a distributed ledger to approve transactions. We provide a brand-new batch-service queuing theory-based end-to-end latency model that, for the first time, defines timers and forks. Additionally, we calculate an analytical estimate of the ideal block size. We demonstrate, supported by simulation findings, that the optimal block size approximation is a reliable technique that achieves nearly ideal performance by drastically lowering the overheads related to blockchain applicationsthat is really fine solution but very expensive which is not suitable so we need to find the lower cost solution regarding transaction rate in blockchain[9].

In 2019 Adiseshu Hari Murali Kodialam T.V. Lakshman the studies Increasing Bitcoin Blockchain Throughput and Latency for High-Throughput Applications A secure distributed ledger that allows for trustworthy transactions between untrusted parties is the Bitcoin blockchain. However, many applications require transaction confirmation rates that are substantially quicker than those of the present Bitcoin blockchain. For expediting Bitcoin's block confirmation process, we introduce ACCEL in this work, a high-throughput, low-latency, deterministic confirmation method. The swift identification of individual blocks that provably belong to the blockchain is the key to our strategy for obtaining speedier confirmation. Singular block detection takes use of the fact that the end-to-end latency between Bitcoin miners is significantly smaller than the inter-block spacing and may be expected to be constrained even when network delays are unbounded. In low-latency, permissioned blockchains, where the block spacing may be tailored to the low latencies of the blockchain to significantly increase throughput, ACCEL is particularly well suited. We assess ACCEL's performance using in-depth simulations as well as an actual implementation that was created with the Bitcoin blockchain in mind and is completely compatible with it. We demonstrate how ACCEL may decrease transaction confirmation latencies to milliseconds with the proper end-to-end latency boundaries, meeting the performance requirements of a variety of applications[10].

2020 will see the completion of S. Alrubei, E. Ball, J. Rigelsford, and C. Willis' article examining the Latency and Performance Analyses of Real-World Wireless IoT-Blockchain Application.In this study, the authors used a flood monitoring and detection system as a real-world IoT-blockchain use case to confirm their results. Additionally, they provide the performance anylsisi, which measures the system end-to-end latency and the transaction arrival time from the node's submission until the transaction arrives on the network. They have shown that, regardless of the connection channel, running their flood detection system, which includes the Ethereum Blockchain Geth client, only uses a modest amount of energy on average.They demonstrated how to incorporate blockchain technology into IoT applications in their study. that permissioned implementation using Ethereum PoA is possible on the IoT. They also draw the conclusion that it is crucial to take the application needs into account, particularly in terms of criticality. When determining the block time and block gas limit to apply, it's also crucial to take the type of communication protocol in use, the quantity of nodes, and their locations into account[11].

The consensus protocol is the assurance for the reliable operation of blockchain systems, according to 2020 research by Shijie Zhang and Jong-Hyouk Lee on the blockchain's primary consensus protocol. Through the use of the consensus protocol, nodes can agree on a certain value or transaction. In this work, they introduced a few well-known blockchain consensus methods and via study and comparison discovered their advantages, disadvantages, and possible use cases. They came to the conclusion that when creating a good consensus protocol, one should take into account not only fault tolerance but also the best way to use it in the right application situation[12].

in 2021 apeh jonathan apeh1 , charles k. ayo2 and ayodele adebiyi3 they worked on improving latency of blockcahin in electronic voting system, They succeeded, and their solution combines a caching database with a web3 API and a lightweight blockchain node known as PUs to reduce latency and make it suitable for massive elections like Nigeria's general elections. In contrast to current systems, their model implements all election procedures, including registration, transmission, tallying/counting, and result visualization[13].

In 2019 fan yang 1 , wei zhou1 , qingqing wu1 , rui long 1 , neal n. xiong They downgraded the delegated evidence of stake and steered clear of it. In order to improve the blockchain's operational efficiency, increase its security, and consume less resources, this article introduces a powerful consensus algorithm called DDPoS. The fundamental concept of this study is to enhance the original DPoS algorithm by combining the benefits of PoW and DPoS. Additionally, they mandate that each node only has one vote for voting at random, which increases fairness, neutralises the right to generate blocks in order to prevent collusion attacks, and enhances node activity across the whole blockchain system.Finally, this study employs a downgrading technique to remove rogue nodes promptly, maintaining the system's security and smooth functioning[14]

In 2019 Ashish Sharma and Dinesh Bhuriya.write The literature review on blockchain in this study discusses how it functions. Another breakthrough that will have a significant impact on how we exchange data and money as a fully developed society is blockchain. Although there is just a small amount of academic research on it since it is so young, it is developing quickly. They began by obtaining a sample from primarily peer-reviewed sources for this writing survey as well as an educational flowchart of articles from various channels. They are able to provide an agent viewpoint on three key areas thanks to their selection of publications. Let's start with some of the most important topics under discussion right now in relation to blockchain innovation. The agent classifications of those points come in second Third, the future of blockchain technology as well as its impact on society and innovation [15].

In 2018 P. S. G. Aruna Sri and D. Lalitha Bhaskari produce a paper describing the blockchain technology study. Blockchain is an emerging technology that allows for the decentralised exchange of transactional data over a vast peer-to-peer network, enabling non-trusting participants to communicate with one another in a verifiable manner without the use of a middleman. In this essay, we go through the fundamentals of blockchain, including its uses, varieties, and mode of operation. The security, privacy, and consensus procedures of this technology are equally significant and cause for worry behind this novel method. This study also discusses the drawbacks of the blockchain technology[16].

In 2020Roman Belfer1, Antonina Kashtalia write about the proof of activity protocol based on network active nodes The new PoA socially oriented network protocol was introduced in this paper. Among its advantages are It reduces monopolisation of resources and power, minimises pseudo-decentralization, and improves network accessibility for each active node.It equitably pays the nodes participating in block formation and blockchain support, correctly chooses node-validators based on effective activity.The PoA protocol can be implemented in any socially oriented setting: social networks, crowdfunding platforms, public sites, and municipal systems. It can be used in all types of organisations where it is simple to determine whether activity is valuable. A new Proof-of-Activity protocol may be used as one of the stages for implementing a new tax policy[17].

The Hybrid Consensus Algorithm Optimisation is a topic that Yaqin Wu, Pengxin Song, and Fuxin Wang write about in 2020. In this essay, they discussed Consensus algorithms have increasingly grown in importance as a result of the continued advancement of blockchain technology. The benefits and drawbacks of a consensus algorithm have an immediate impact on how well a blockchain performs and operates. Low latency, high throughput, strong scalability, and decentralisation are all desirable characteristics in a consensus method. This paper proposes an improved hybrid consensus algorithm based on the PBFT algorithm and the POS algorithm as a replacement for the current consensus method. It uses verifiable cryptographic sortition to dynamically choose the consensus node. which guarantees the low latency and high throughput of the consensus process in addition to allowing a large number of nodes to participate in the consensus equitably. In order to reduce block forking, we will continue to research the blockchain consensus algorithm and make improvements to it in the future. To prevent the performance loss brought on by the ledger forking, the blockchain ledger structure will be improved and optimised, and the ledger will be rebuilt as a directed acyclic graph. This will enhance the functionality of the blockchain consensus mechanism.[18].

In 2022 Ke Wang and Hyong S. Kim write a paper about the consensus latency of proof of Work In this study, they examine the Proof of Work (PoW) blockchains' consensus delay. A block is considered to have reached consensus if it is a part of every miner's longest chain. In this study, we define consensus latency as the time elapsed between the mining of a block and the block's initial consensus. We can decide when to confirm blocks for a number of different confirmation criteria by using knowledge about consensus latency. They discover that in real-world situations, blocks might quickly attain consensus. For instance, a block might achieve consensus in an average of 1.3 deltas on a propagation network with the maximum network delay delta. Contrary to popular belief, average consensus latency actually reduces as mining complexity increases. In order to comprehend the relationship between consensus latency and the analytical model we construct[19].

CHAPTER 3

Design and Methodology

3.1. Overview

This section presents the road map of the proposed approach. The design entails the choice of pertinent characteristics, model selection, and validation. The methodology describes the steps in exploration of consensus protocol , analytical modeling ,Mapping of analytical modeling to sequence diagram ,Calculating the latency on on blockchain model,simulating results and validating the results. The strategy entails creating analytical models to describe important performance traits, translating these models to sequence diagrams for visual representation, and measuring real-based latency to gauge usable efficiency. The effectiveness of the methods is investigated using simulations, and the outcomes are thoroughly confirmed against testing enviourrment or test bed tests. This in-depth study intends to provide useful insights into the protocols' scalability, decentralization, performance, security, and other aspects, directing future advancements in consensus protocol design and implementation for more effective and safe blockchain networks

3.2. Proposed Methodology

This research's suggested technique for the critical study of consensus protocols intends to systematically assess and contrast several blockchain consensus mechanisms, including PoW, PoS, DPoS, and PBFT. It entails creating analytical models, translating them to sequence diagrams, figuring out real-based delay, running simulations, and verifying outcomes using data from the actual world. This all-encompassing strategy aims to offer insightful information on the effectiveness and performance of consensus protocols, leading further development of blockchain technology.

To accomplish the objectives, the proposed approach is segmented into five phases:

- Phase 1: Exploration of consensus protocol,
- Phase 2: Analytical modeling
- Phase 3: Mapping of analytical modeling to sequence diagram
- Phase 4: Design Blockchain Model for validating the analytical model
- Phase 5: Simulating the results

The flow of these five phases is represented in Figure 3.2 Each phase is separately described in detail in this section.

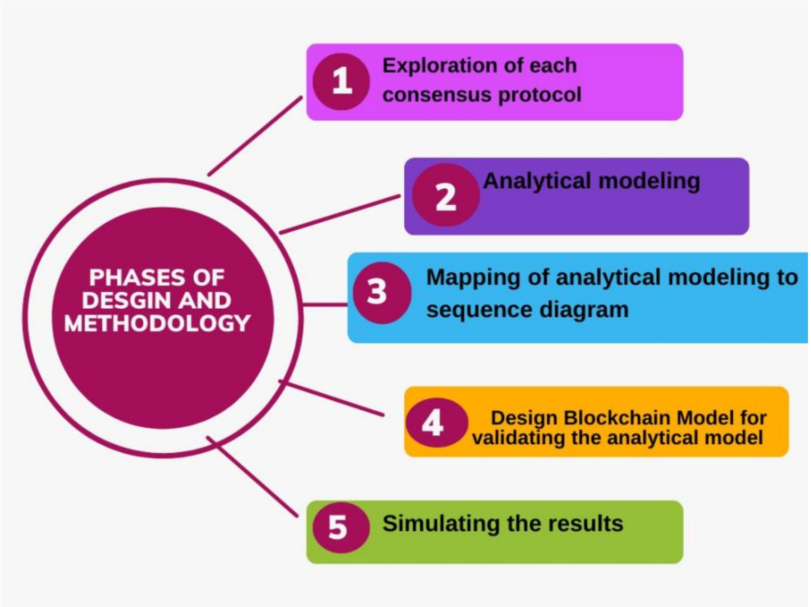

PHASES OF DESGIN AND METHODOLOGY
1
Exploration of each consensus protocol
2
Analytical modeling
3
Mapping of analytical modeling to sequence diagram
4
Design Blockchain Model for validating the analytical model
5
Simulating the results

Figure 3.2 five phases covering the complete methodology proposed for analysis of consensus protocol.

PHASE 1

3.2.1.Exploration of each Consensus Protocol

3.2.1.1.Proof of Work (PoW)

Blockchain networks employ the Proof of Work (PoW) consensus process, in which users, known as miners, solve computationally difficult problems to add new blocks and confirm transactions. This technique uses a lot of computer resources and energy, yet the answer is simple to check with others. The longest valid chain with the greatest cumulative computing labour becomes the "true" blockchain once miners are paid for their efforts.

3.2.1.2.Proof of Stake (PoS)

Blockchain networks employ the Proof of Stake (PoS) consensus method, where validators are chosen based on the amount of bitcoin they "stake" in the network. Validators propose and validate new blocks alternately, and their chances of selection are based on their stake. PoS is more energy-efficient than Proof of Work (PoW) since it does not need mining or computational problems, both of which consume a lot of energy. Additionally, it may benefit from scalability. However, to guarantee the security and integrity of the network, issues like the "nothing-at-stake" dilemma must be resolved. PoS, in general, offers a different method for reaching agreement by relying on members' stake in the network rather than processing capacity.

3.2.1.3. Delegated Proof of Stake(DPOS)

It is a variant of Proof of Stake that focuses on the voting process and reputation. A network node is chosen as a delegated node if and only if its reputation is higher and all other nodes support it.A user's vote will have greater weight when electing the delegate if they have more coins in their possession.

3.2.1.4. Proof of Importance (PoI)

Each node in the network is given a significant score by this algorithm. The relevance score is based on the quantity of tokens owned, network activity, reputation, and the number of transactions performed to and from the specific account, among other things.

3.2.1.5. Proof of Authority (PoA)

A consensus technique called Proof of Authority (PoA) allows for trustworthy parties to confirm transactions and build blocks based on their reputation and identity. Although it supports quick transaction processing, it is more suited for private or consortium blockchains than public networks since it depends on a centralised set of validators.

3.2.1.6. Proof of elapsed time (PoET)

A blockchain network's participants fight for a randomly allocated wait time using the Proof of Elapsed Time (PoET) consensus method, and the participant with the shortest wait time is selected as the block creator. PoET saves energy and does away with the necessity for computationally demanding activities. It offers a fair and secure consensus process and is widely utilised in permissioned blockchain networks.

3.2.1.7. Proof of Burn (PoB)

In order to demonstrate their commitment and compete for the opportunity to validate transactions, Proof of Burn (PoB) participants burn their cryptocurrency tokens. It provides an alternative to established consensus techniques like PoW and PoS, but is less popular and is frequently employed in niche blockchain applications.

3.2.1.8. Proof of Capacity / Proof of Space (PoC / PoSpace)

Participants in a blockchain provide storage space to mine or validate blocks as part of the Proof of Capacity (PoC) or Proof of Space (PoSpace) consensus method. Participants compete to solve riddles by creating plots loaded with pre-computed data; the quickest answer earns the privilege to mine and incentives. PoC is less popular than more established techniques like Proof of Work (PoW) or Proof of Stake (PoS), despite being energy-efficient and suited for networks with surplus storage capacity.

3.2.1.9. Proof of History (PoH)

The Solana blockchain network created the timekeeping method known as Proof of History (PoH). By giving each event a distinctive evidence or timestamp, it creates a verifiable and unchangeable ordering of events in a decentralized network. As a result, Solana is able to implement effective consensus methods, achieve high levels of scalability and transaction throughput, and maintain a trustworthy and secure blockchain network

3.2.1.10. Proof of Activity

A hybrid consensus approach called Proof of Activity (PoA) combines Proof of Work (PoW) with Proof of Stake (PoS). In a PoW system, miners compete to produce new blocks, which are then verified in a PoS system by proving ownership of a stake in the network. To make PoA an effective and reliable consensus for blockchain networks, it attempts to lower energy consumption while ensuring security and decentralization

PHASE2

3.3 Analytical Modeling

3.2.2. By using mathematical equations and parameters, analytical modelling is utilised to calculate the latency of a consensus protocol in a blockchain network. It enables researchers to evaluate aspects like propagation speed, block size, and network circumstances to spot bottlenecks and improve the protocol's architecture. This method assists in helping to improve the effectiveness and dependability of consensus processes in practical blockchain applications. So we drive Equation from analytical modeling for each consensus protocol.

3.3.1.Proof of work (PoW)

Certainly! Here are the step-by-step mathematical calculations to determine the latency of proof-of-work:

Step 1: Gather the necessary information:

- Difficulty (D): The current difficulty level of the proof-of-work algorithm.
- Hashrate (H): The computational power of the network in hashes per second.
- Block Time (B): The average time taken to mine a new block.

Step 2: Convert the difficulty to a target value:

- Target (T): Divide the maximum target value (which is a constant specific to the proof-of-work algorithm) by the difficulty.

T = Maximum Target / D

Step 3: Calculate the number of hashes required to find a valid solution:

- Hashes (N): Divide the target value by the probability of finding a valid solution with each hash attempt.

N = T / (2^256)

Step 4: Determine the time taken to perform the required number of hashes:

- Hash Time (Ht): Divide the total number of hashes (N) by the hashrate (H) to get the time taken for the required number of hashes.

Ht = N / H

Step 5: Calculate the time taken to mine a block:

- Block Time (Bt): Multiply the block time (B) by the number of required hashes per block (N).

Bt = B * N

Step 6: Compute the latency or average time to find a valid solution:

- Latency (L): Add the hash time (Ht) and block time (Bt) to get the total latency.

L = Ht + Bt

By following these steps and plugging in the appropriate values for Difficulty, Hashrate, and Block Time, you can calculate the latency of proof-of-work. The resulting value will represent the average time it takes to find a valid solution and mine a new block in the blockchain.

Table 3.3.1

Units of each attribute to calculate the latency in proof of work are illustrate

Feature	**Units**
Difficulty (D)	Unitless (difficulty level).
Hashrate (H)	Hashes per second (H/s).
Block Time (B)	Seconds (s)
Target (T)	Unitless (target value)
Hashes (N)	Unitless (number of hashes).
Hash Time (Ht)	Seconds (s).

Block Time (Bt)	Seconds (s).
Latency (L)	Seconds (s)

In this calculation, most of the values are unitless, except for the hashrate (H), block time (B), hash time (Ht), block time (Bt), and latency (L), all of which are measured in seconds (s). The unitless values in the calculation are ratios or constants that are used to derive other values, so they don't have specific units associated with them.

3.3.2.Proof of Stake (PoS)

Certainly! Here are the step-by-step mathematical calculations to determine the latency of proof-of-stake:

Step 1: Gather the necessary information:

- Total Supply (S): The total supply of tokens in the proof-of-stake blockchain.
- Active Stake (A): The total number of tokens actively staked by validators.
- Slot Time (T): The average time duration of each slot in the proof-of-stake protocol.

Step 2: Calculate the stake participation ratio:

- Stake Participation Ratio (P): Divide the active stake (A) by the total supply (S).

$$P = A / S$$

Step 3: Determine the average time to generate a block:

- Block Time (Bt): Multiply the stake participation ratio (P) by the slot time (T) to get the average time taken to generate a block.

$$Bt = P * T$$

Step 4: Calculate the latency or average time to finalize a block:

- Latency (L): Add the block time (Bt) to the slot time (T) to get the total latency.

$$L = Bt + T$$

By following these steps and plugging in the appropriate values for Total Supply, Active Stake, and Slot Time, you can calculate the latency of proof-of-stake. The resulting value will represent the average time it takes to finalize a block in the proof-of-stake blockchain.

Table 3.3.2

Units of each attribute to calculate the latency in proof of Stake are illustrate

Feature	**Units**
Total Supply (S)	Tokens
Active Stake (A)	Tokens
Slot Time (T)	Time (seconds, minutes, etc.)
Stake Participation Ratio (P)	Unitless (ratio of tokens).
Block Time (Bt)	Time (seconds, minutes)
Latency (L)	Seconds (s).

In this calculation, the units are as follows:

Total Supply (S) and Active Stake (A) are both measured in tokens.

Slot Time (T) is measured in time units (seconds, minutes, etc.).

The Stake Participation Ratio (P) is unitless, as it's a ratio of token amounts.

The Block Time (Bt) and Latency (L) are both measured in the same time units as the Slot Time (T), whatever time unit you are using

3.3.3.Delegated proof of Stake (DPOS)

Certainly! Here are the step-by-step mathematical calculations to determine the latency of delegated proof-of-stake (DPoS):

Step 1: Gather the necessary information:

- Total Supply (S): The total supply of tokens in the DPoS blockchain.
- Active Stake (A): The total number of tokens actively staked by all validators.
- Delegation Ratio (D): The ratio of tokens delegated to a specific validator.
- Block Time (Bt): The average time duration of each block in the DPoS protocol.
- Number of Validators (N): The total number of validators in the DPoS network.

Step 2: Calculate the stake participation ratio:

- Stake Participation Ratio (P): Divide the active stake (A) by the total supply (S).

P = A / S

Step 3: Calculate the individual validator's probability of being selected:

- Validator Probability (Vp): Divide the delegation ratio (D) by the sum of all delegation ratios across all validators.

Vp = D / (Sum of all Delegation Ratios)

Step 4: Calculate the average time to generate a block by a specific validator:

- Validator Block Time (Vbt): Multiply the validator probability (Vp) by the block time (Bt).

Vbt = Vp * Bt

Step 5: Calculate the average time for a specific validator to produce a block:

- Validator Production Time (Vpt): Divide the total number of validators (N) by the number of validators selected per block (which is a constant specific to the DPoS protocol).

Vpt = 1 / (N / Validators Selected per Block)

Step 6: Calculate the latency or average time to finalize a block:

- Latency (L): Multiply the validator production time (Vpt) by the validator block time (Vbt).

L = Vpt * Vbt

By following these steps and plugging in the appropriate values for Total Supply, Active Stake, Delegation Ratio, Block Time, and Number of Validators, you can calculate the latency of delegated proof-of-stake. The resulting value will represent the average time it takes to finalize a block in the DPoS blockchain

Table 3.3.3

Units of each attribute to calculate the latency in Delegated Proof of Stake are illustrate

Feature	**Units**
Total Supply (S)	Tokens
Active Stake (A)	Tokens
Delegation Ratio (D)	Ratio(Unitless)
Block Time (Bt)	Seconds (s)
Number of Validators (N)	Count(Unitless)
Stake Participation Ratio (P)	Unitless(Ratio of Tokens)
Validator Probability (Vp)	Unitless (ratio).
Validator Block Time (Vbt)	Time (same units as Block Time).
Validator Production Time (Vpt)	Time (seconds, minutes, etc.).
Latency (L)	Time (same units as Block Time)

In summary, the units are as follows:

Total Supply (S) and Active Stake (A) are both measured in tokens.

Delegation Ratio (D) is unitless, as it's a ratio.

Block Time (Bt), Validator Block Time (Vbt), Validator Production Time (Vpt), and Latency (L) are all measured in the same time units (seconds, minutes, etc.).

Number of Validators (N) is a count, so it's unitless.

It's crucial to maintain consistent units throughout the calculations for accurate results.

3.3.4.Proof of Importance (PoI)

Step 1: Gather the necessary information:

Start Timestamp (St): The timestamp when a transaction is initiated or entered into the network.

Confirmation Timestamp (Ct): The timestamp when the transaction is confirmed or included in a block.

Step 2: Calculate the latency:

Latency (L): Subtract the start timestamp (St) from the confirmation timestamp (Ct) to determine the elapsed time or latency. L = Ct - St

The resulting value, Latency (L), represents the estimated time it takes for a transaction to be confirmed in the Proof of Importance mechanism.

Table 3.3.4

Units of each attribute to calculate the latency in Proof of Importance are illustrate

Feature	**Units**
Start Timestamp (St):	Time (timestamp with units like seconds, milliseconds, etc.).
Confirmation Timestamp (Ct)	Time (timestamp with the same units as Start Timestamp)
Latency (L)	Time (same units as Start and Confirmation Timestamps).

In this context, both the Start Timestamp (St) and Confirmation Timestamp (Ct) are measured in time units, such as seconds or milliseconds. The calculated Latency (L) will also be measured in the same time units as the Start and Confirmation Timestamps, representing the time interval between the initiation and confirmation of the transaction.

3.3.5.Proofof Authority(PoA)

To calculate the latency in a Proof of Authority (PoA) consensus algorithm, which refers to the time it takes for a newly created block to be confirmed or finalized, you can follow these steps:

Step 1: Gather the necessary information:

- Block Creation Time (Bct): The time it takes for a block to be created by the designated authorities.

- Block Confirmation Time (Bcn): The time it takes for the block to be confirmed by the authorities in the network.

Step 2: Calculate the latency:

- Latency (L): Add the block creation time (Bct) and the block confirmation time (Bcn) to determine the total time it takes for a newly created block to be confirmed.

$$L = Bct + Bcn$$

The resulting value, Latency (L), represents the estimated time it takes for a newly created block to be confirmed in the PoA consensus algorithm.

It's important to note that the specific values for block creation time and block confirmation time can vary depending on the PoA implementation and network conditions. The block creation time is determined by the authorities responsible for creating the blocks, while the block confirmation time depends on the communication and consensus process among the authorities

Table 3.3.5

Units of each attribute to calculate the latency in Proof of Authority are illustrate

Feature	**Units**
Block Creation Time (Bct)	Time (seconds, milliseconds, etc.).
Block Confirmation Time (Bcn)	Time (seconds, milliseconds, etc.).
Latency (L)	Time (seconds, milliseconds, etc.).

In this context:

Block Creation Time (Bct) and Block Confirmation Time (Bcn) are both measured in time units, such as seconds or milliseconds.

The calculated Latency (L) will also be measured in the same time units as Block Creation Time and Block Confirmation Time.

3.3.6.Proof of elapsed time (PoET)

Certainly! Here are the mathematical steps to calculate the latency of proof elapsed time in a Proof of History (PoH) mechanism:

Step 1: Gather the necessary information:

- Start Timestamp (St): The initial timestamp in the PoH sequence.
- End Timestamp (Et): The final timestamp in the PoH sequence.

Step 2: Calculate the proof elapsed time:

- Proof Elapsed Time (PET): Subtract the start timestamp (St) from the end timestamp (Et) to determine the elapsed time.

PET = Et - St

Step 3: Gather additional information:

- Block Time (Bt): The average time duration between consecutive blocks in the PoH mechanism.

- Block Confirmation Threshold (Ct): The number of block confirmations required for a block to be considered finalized.

Step 4: Calculate the latency of proof elapsed time:

- Latency (L): Multiply the proof elapsed time (PET) by the block confirmation threshold (Ct) and the block time (Bt).

L = PET * Ct * Bt

The resulting value, Latency (L), represents the estimated time it takes for the proof elapsed time to be confirmed by the specified number of block confirmations in the PoH mechanism, considering the block time.

Table 3.3.6

Units of each attribute to calculate the latency in Proof of elapsed time (PoET)areillustrate

Feature	**Units**
Start Timestamp (St)	Time (timestamp seconds, milliseconds)
End Timestamp (Et)	Time (timestamp seconds, milliseconds)
Proof Elapsed Time (PET):	Time (timestamp seconds, milliseconds)
Block Time (Bt)	Time (seconds, millisecond)
Block Confirmation Threshold (Ct)	Count (unitless)
Latency (L)	Time (same units as Block Time)

3.3.7.Proof of Burn (PoB)

Certainly! Here are the mathematical steps to calculate the latency of a simplified Proof of Burn mechanism:

Step 1: Gather the necessary information:

- Amount to Burn (B): The number or value of tokens a participant intends to burn.
- Burning Rate (R): The rate at which tokens are burned, typically expressed as tokens per unit of time.

Step 2: Calculate the time required to complete the burning process:

- Burn Time (T): Divide the amount to burn (B) by the burning rate (R).

$$T = B / R$$

The resulting value, Burn Time (T), represents the estimated time it would take for the participant to complete the burning process based on the burn rate and the desired amount to burn.

Table 3.3.7

Units of each attribute to calculate the latency in Proof of Burn are illustrate

Feature	**Units**
Amount to Burn (B)	Tokens or a value (units depend on the specific cryptocurrency or asset).
Burning Rate (R)	Tokens per unit of time (units could be tokens per second, per minute)
Burn Time (T)	Time (seconds, minutes, etc.).

In this context:

Amount to Burn (B) can have units depending on the specific cryptocurrency or asset (e.g., tokens).

Burning Rate (R) is measured in tokens per unit of time (e.g., tokens per second).

Burn Time (T) will be measured in the same time units as the units used for the Burning Rate (R).

3.3.8. Proof of Capacity / Proof of Space (PoC / PoSpace)

Calculating the latency of Proof of Space involves determining the time it takes to generate a valid proof based on the allocated space. However, Proof of Space typically does not have a fixed latency since it depends on the specific algorithm and implementation. Still, I can provide you with a general outline of the steps involved:

Step 1: Gather the necessary information:

Allocated Space (AS): The amount of storage space allocated by the participant for Proof of Space.

Difficulty (D): The difficulty level of the Proof of Space algorithm.

Step 2: Determine the number of hashes required:

Hashes (H): Multiply the allocated space (AS) by the difficulty (D).

$H = AS * D$

Step 3: Calculate the time taken to perform the required number of hashes:

Hash Time (Ht): Divide the total number of hashes (H) by the computational power or speed of the hardware used for hashing.

$Ht = H / \text{Hashing Speed}$

Step 4: Compute the latency or average time to generate a valid proof:

Latency (L): The time taken to perform the required number of hashes (Ht) represents the latency or average time to generate a valid proof.

It's important to note that Proof of Space algorithms vary in terms of how they utilize storage space and generate proofs. Therefore, the specific details of calculating latency may differ based on the chosen Proof of Space

Table 3.3.8

Units of each attribute to calculate the latency in Proof of Space are illustrate

Feature	**Units**
Allocated Space (AS)	Storage space (units depend on the chosen unit of storage, like gigabytes, terabytes)
Difficulty (D)	Unitless (difficulty level)
Hashes (H)	Unitless (number of hashes)
Hash Time (Ht)	Time (seconds, minutes)
Latency (L)	Time (same units as Hash Time)

In this context:

Allocated Space (AS) is measured in storage units, such as gigabytes, terabytes, etc.

Difficulty (D) is unitless, representing a difficulty level.

Hashes (H) are unitless, representing the number of hashes required.

Hash Time (Ht) is measured in time units, such as seconds or minutes.

The calculated Latency (L) will be measured in the same time units as Hash Time.

3.3.9.Proof of History (PoH)

general outline of the steps involved in estimating the latency of PoH:

Step 1: Gather the necessary information:

Start Timestamp (St): The timestamp when a particular event or transaction occurred.

End Timestamp (Et): The timestamp when the event or transaction is confirmed or finalized.

Step 2: Calculate the latency:

Latency (L): Subtract the start timestamp (St) from the end timestamp (Et) to determine the elapsed time or latency. L = Et - St

The resulting value, Latency (L), represents the estimated time it takes for an event or transaction to be confirmed or finalized in the Proof of History mechanism.

Please note that the specific calculations and units of time can vary depending on the implementation details and design choices of the PoH mechanism. Additionally, PoH can be used in conjunction with other consensus mechanisms, which can further influence the overall latency in a blockchain network.

Table 3.3.9

Units of each attribute to calculate the latency in Proof of History are illustrate

Feature	**Units**
Start Timestamp (St)	Time (timestamp with units like seconds, milliseconds)
End Timestamp (Et)	Time (timestamp with units like seconds, milliseconds)
Latency (L)	Time (same units as Start and End Timestamps).

In this context:

Start Timestamp (St) and End Timestamp (Et) are both measured in time units, such as seconds or milliseconds.

The calculated Latency (L) will also be measured in the same time units as Start and End Timestamps, representing the time interval between the event or transaction occurrence and its confirmation or finalization.

3.3.10. Proof of Activity

if you're referring to a hypothetical concept where both Proof of Work (PoW) and Proof of Stake (PoS) are combined, you can consider the following steps:

Step 1: Gather the necessary information:

- Hashrate (H): The computational power of the network in hashes per second.
- Difficulty (D): The difficulty level of the Proof of Work algorithm.
- Stake (S): The total amount of tokens staked in the Proof of Stake component.
- Block Time (Bt): The average time duration between consecutive blocks.

Step 2: Calculate the Proof of Work (PoW) component's latency:

- PoW Latency (PwL): Use the same mathematical steps as described earlier for calculating the latency of Proof of Work. This involves considering the difficulty, hashrate, and block time specific to the PoW component.

Step 3: Calculate the Proof of Stake (PoS) component's latency:

- PoS Latency (PsL): Use the same mathematical steps as described earlier for calculating the latency of Proof of Stake. This typically involves considering the stake participation ratio, block time, and the number of validators specific to the PoS component.

Step 4: Combine the latencies of both components:

- Total Latency (TL): Combine the PoW latency (PwL) and PoS latency (PsL) to get the overall latency of the Proof of Activity mechanism.

TL = PwL + PsL

Table 3.10

Units of each attribute to calculate the latency in Proof of are Activity illustrate

Feature	**Units**
Hashrate (H)	Hashes per second (H/s)
Difficulty (D)	Unitless (difficulty level).
Stake (S)	Tokens or a value (units depend on the specific cryptocurrency or asset)
Block Time (Bt)	Time (seconds, minutes, etc.)
PoW Latency (PwL)	Time (same units as Block Time).
PoS Latency (PsL)	Time (same units as Block Time).
Total Latency (TL)	Time (same units as Block Time).

In this context:

Hashrate (H) is measured in hashes per second (H/s).

Difficulty (D) is unitless, representing a difficulty level.

Stake (S) is measured in tokens or a specific unit, depending on the cryptocurrency or asset.

Block Time (Bt) is measured in time units, such as seconds or minutes.

The calculated PoW Latency (PwL), PoS Latency (PsL), and Total Latency (TL) will all be measured in the same time units as Block Time.

PHASE 3

3.4 Mapping of analytical modeling to sequence diagram

3.3.11. Proof of Work

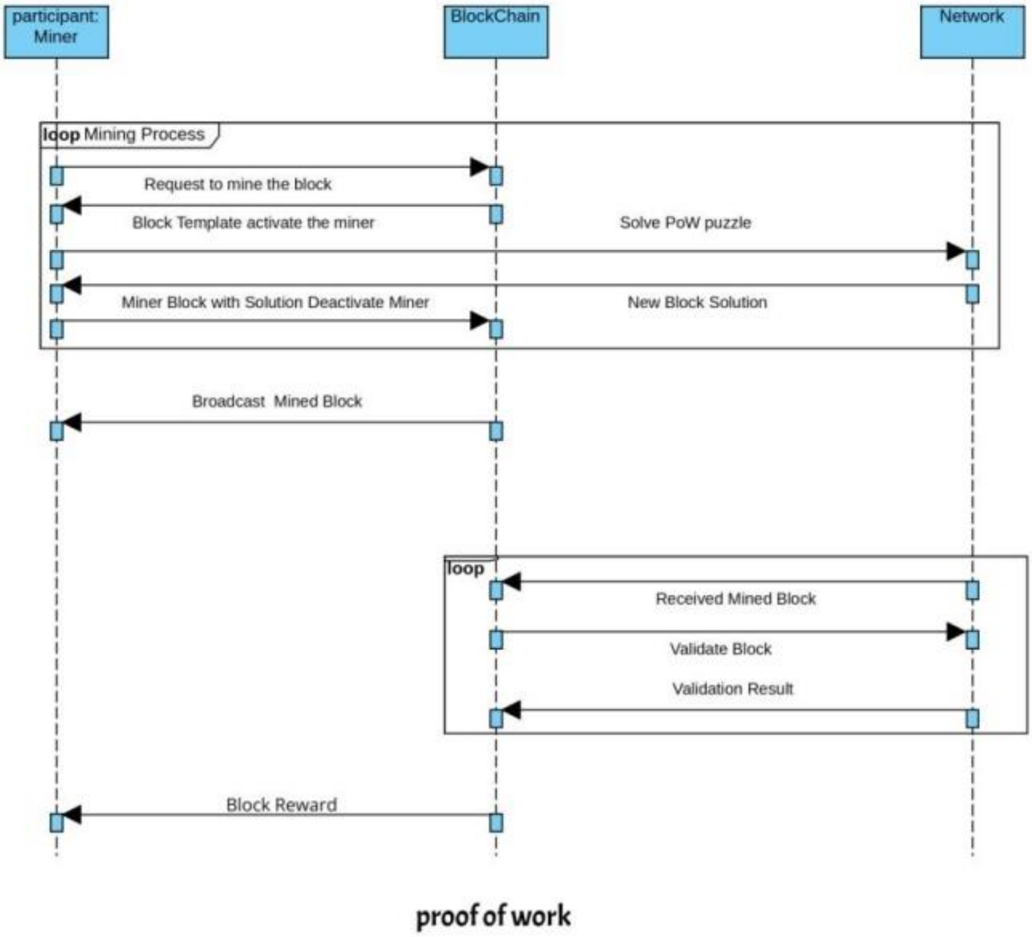


Figure 3.4.1 Proof of Work

One to one mapping of sequence diagram to analytical modeling of Proof of Work

participant Miner

participant Blockchain

participant Network

loop Mining Process

```
    Miner -> Blockchain: Request to Mine Block
    Blockchain -> Miner: Block Template
    activate Miner
    Miner -> Network: Solve PoW Puzzle
    Network -> Miner: New Block Solution
    Miner -> Blockchain: Mined Block with Solution
    deactivate Miner
end

Blockchain -> Network: Broadcast Mined Block

loop Block Validation
    Network -> Blockchain: Received Mined Block
    Blockchain -> Network: Validate Block
    Network -> Blockchain: Validation Result
end

Blockchain -> Miner: Block Reward

loop Latency Calculation
    Miner -> Blockchain: Gather Necessary Information
    Blockchain -> Miner: Difficulty (D), Hashrate (H), Block Time (B)
    activate Miner
    Miner -> Miner: Convert Difficulty to Target (T)
    Miner -> Miner: Calculate Number of Hashes (N)
    Miner -> Miner: Determine Hash Time (Ht)
    Miner -> Miner: Calculate Block Time (Bt)
    Miner -> Miner: Compute Latency (L)
    deactivate Miner
 end
```

After the mining process loop, there is a new loop called "Latency Calculation" to incorporate the analytical modeling equations into the sequence diagram.

The Miner gathers the necessary information (Difficulty, Hashrate, and Block Time) from the Blockchain for latency calculation.

The Miner then performs the following calculations sequentially within the loop:

Convert Difficulty to Target (T)

Calculate Number of Hashes (N)

Determine Hash Time (Ht)

Calculate Block Time (Bt)

Compute Latency (L)

The result of the latency calculation (L) is not explicitly shown in the sequence diagram, but it can be used for performance analysis or other purposes.

By adding this "Latency Calculation" loop, we integrate the analytical modeling equations into the sequence diagram, demonstrating how the latency of Proof of Work is computed based on the given variables and formulas.

3.3.12. Proof of Stake

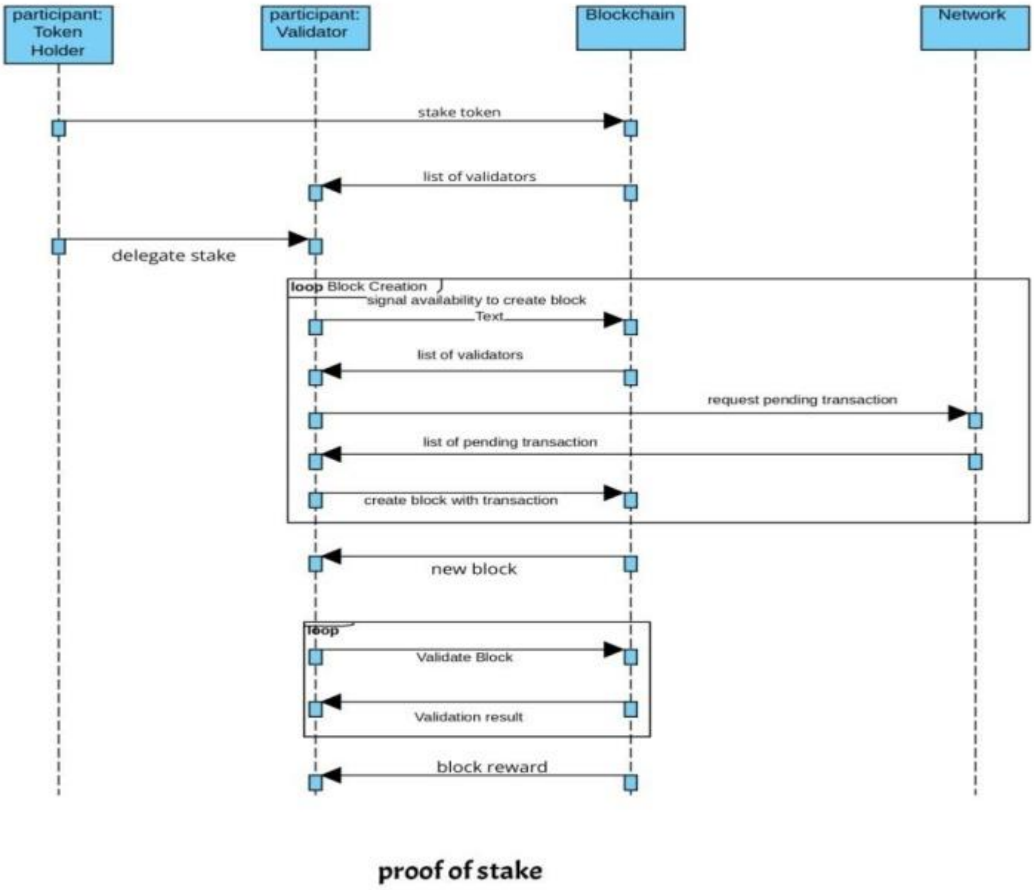


Figure 3.4.2 Proof of Stake

One to one mapping of sequence diagram to analytical modeling of Proof of Stake

participant Token Holder

participant Validator

participant Blockchain

participant Network

Token Holder -> Blockchain: Stake Tokens

Blockchain -> Validator: List of Validators

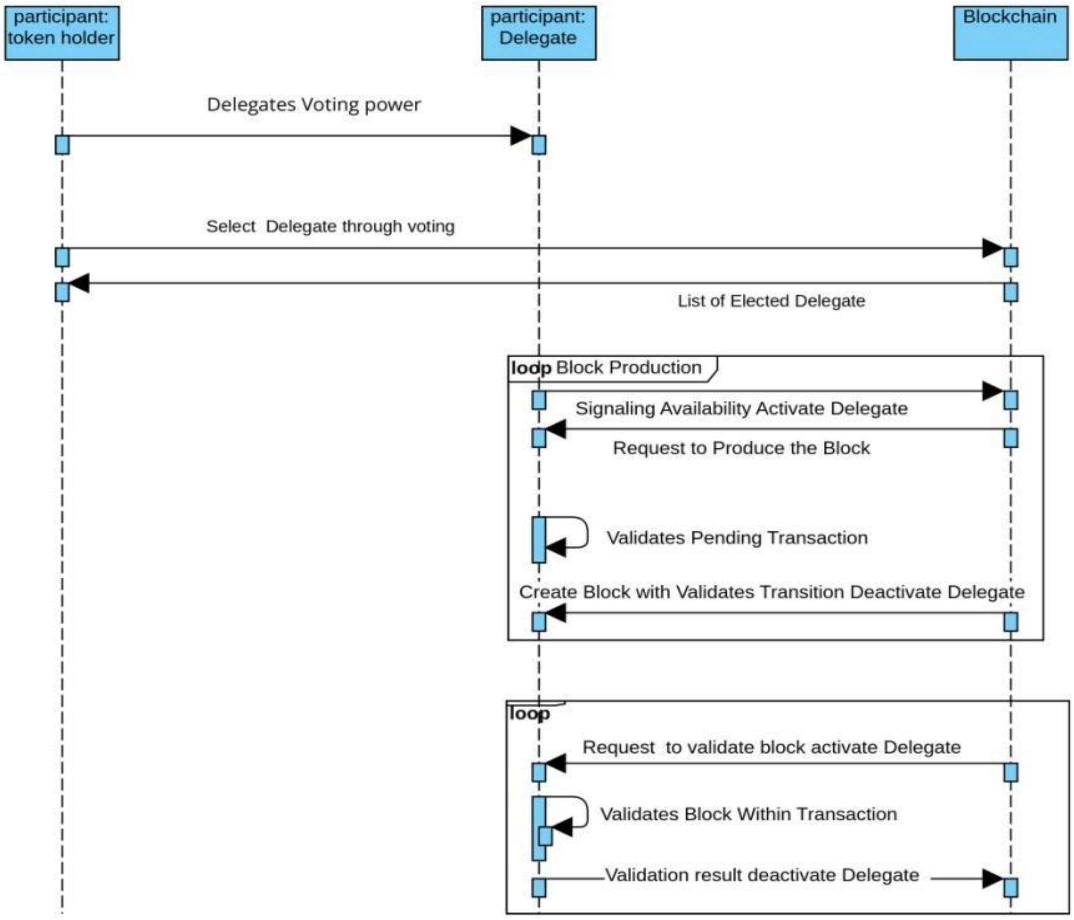
Token Holder -> Validator: Delegate Stake

```
loop Block Creation
    Validator -> Blockchain: Signal Availability to Create Block
    Blockchain -> Validator: List of Validators
    Validator -> Network: Request Pending Transactions
    Network -> Validator: List of Pending Transactions
    Validator -> Blockchain: Create Block with Transactions
end

Blockchain -> Validator: New Block

loop Block Validation
    Validator -> Blockchain: Validate Block
    Blockchain -> Validator: Validation Result
end

Blockchain -> Validator: Block Reward

loop Latency Calculation
    Validator -> Blockchain: Gather Necessary Information
    Blockchain -> Validator: Total Supply (S), Active Stake (A), Slot Time (T)
    activate Validator
    Validator -> Validator: Calculate Stake Participation Ratio (P)
    Validator -> Validator: Determine Block Time (Bt)
    Validator -> Validator: Compute Latency (L)
    deactivate Validator
```

end

- After the "Block Validation" loop, there is a new loop called "Latency Calculation" to incorporate the analytical modeling equation into the sequence diagram.
- The Validator gathers the necessary information (Total Supply, Active Stake, and Slot Time) from the Blockchain for latency calculation.
- The Validator then performs the following calculations sequentially within the loop:
- Calculate Stake Participation Ratio (P): P = A / S
- Determine Block Time (Bt): Bt = P * T
- Compute Latency (L): L = Bt + T
- The result of the latency calculation (L) is not explicitly shown in the sequence diagram, but it represents the average time it takes to finalize a block in the Proof of Stake (PoS) blockchain.
- By adding this "Latency Calculation" loop, we integrate the analytical modeling equation into the sequence diagram, demonstrating how the latency of Proof of Stake can be calculated based on the given variables and formulas.

3.3.13. Delegated Proof of Stake


participant: token holder
participant: Delegate
Blockchain
Delegates Voting power
Select Delegate through voting
List of Elected Delegate
loop Block Production
Signaling Availability Activate Delegate
Request to Produce the Block
Validates Pending Transaction
Create Block with Validates Transition Deactivate Delegate
loop
Request to validate block activate Delegate
Validates Block Within Transaction
Validation result deactivate Delegate

Delegated proof of stake

Figure 3.4.1 Delegated Proof of Stake

One to one mapping of sequence diagram to analytical modeling of Delegated Proof of Work

participant Token Holder

participant Delegate

participant Blockchain

Token Holder -> Blockchain: Selects Delegates through Voting

Blockchain -> Delegate: List of Elected Delegates

```
Token Holder -> Delegate: Delegates Voting Power

loop Block Production
    Delegate -> Blockchain: Signaling Availability
    activate Delegate
    Blockchain -> Delegate: Request to Produce Block
    Delegate -> Delegate: Validates Pending Transactions
    Delegate -> Blockchain: Created Block with Validated Transactions
    deactivate Delegate
end

loop Block Validation
    Blockchain -> Delegate: Request to Validate Block
    activate Delegate
    Delegate -> Delegate: Validates Transactions within Block
    Delegate -> Blockchain: Validation Result
    deactivate Delegate
end

Blockchain -> Token Holder: Block Confirmation
Token Holder -> Blockchain: Stake Delegation

loop Latency Calculation
    Delegate -> Blockchain: Gather Necessary Information
    Blockchain -> Delegate: Total Supply (S), Active Stake (A), Delegation Ratio (D),
Block Time (Bt), Number of Validators (N)
    activate Delegate
    Delegate -> Delegate: Calculate Stake Participation Ratio (P)
    Delegate -> Delegate: Calculate Validator Probability (Vp)
    Delegate -> Delegate: Calculate Validator Block Time (Vbt)
    Delegate -> Delegate: Calculate Validator Production Time (Vpt)
    Delegate -> Delegate: Compute Latency (L)
    deactivate Delegate
end
```

- In this updated sequence diagram:
- After the "Block Validation" loop, there is a new loop called "Latency Calculation" to incorporate the analytical modeling equation into the sequence diagram.
- The Delegate gathers the necessary information (Total Supply, Active Stake, Delegation Ratio, Block Time, and Number of Validators) from the Blockchain for latency calculation.
- The Delegate then performs the following calculations sequentially within the loop:
- Calculate Stake Participation Ratio (P): P = A / S
- Calculate Validator Probability (Vp): Vp = D / (Sum of all Delegation Ratios)
- Calculate Validator Block Time (Vbt): Vbt = Vp * Bt
- Calculate Validator Production Time (Vpt): Vpt = 1 / (N / Validators Selected per Block)
- Compute Latency (L): L = Vpt * Vbt
- The result of the latency calculation (L) is not explicitly shown in the sequence diagram, but it represents the average time it takes to finalize a block in the Delegated

Proof of Stake (DPoS) blockchain.

- By adding this "Latency Calculation" loop, we integrate the analytical modeling equation into the sequence diagram, demonstrating how the latency of DPoS can be calculated based on the given variables and formulas.

3.3.14. Proof of Importance (PoI)

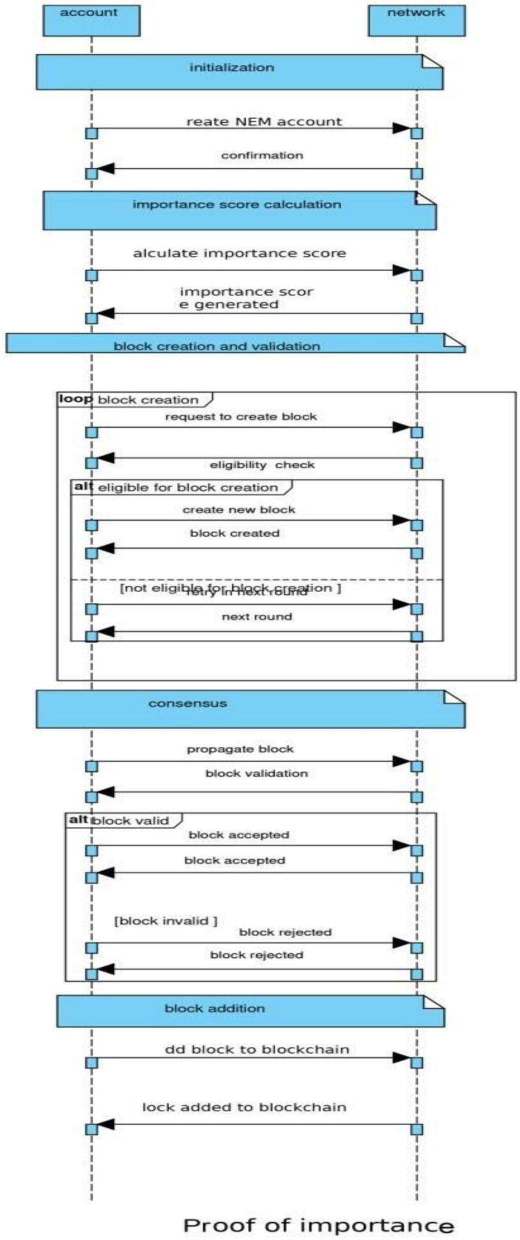


Figure 3.4.4 proof of importance

One to one mapping of sequence diagram to analytical modeling of Proof of

Importance

```
participant Node

participant Blockchain

participant Network


Node -> Blockchain: Stake Tokens

Blockchain -> Node: Confirmation of Staked Tokens

Node -> Blockchain: Participate in Network Activity


loop PoI Calculation

    Node -> Node: Calculate Reputation Score

    Node -> Node: Calculate Activity Metrics

    Node -> Node: Calculate Importance Score

end


loop Block Creation

    Node -> Blockchain: Signal Availability to Create Block

    Blockchain -> Node: Selected as Block Creator

    Node -> Network: Request Pending Transactions

    Network -> Node: List of Pending Transactions

    Node -> Blockchain: Create Block with Transactions

end


Blockchain -> Node: New Block
```

```
loop Block Validation
    Network -> Blockchain: Received Mined Block
    Blockchain -> Node: Request to Validate Block
    Node -> Node: Validate Transactions within Block
    Node -> Blockchain: Validation Result
end

Blockchain -> Node: Block Reward

loop Transaction Latency Calculation
    Node -> Node: Gather Necessary Information
    Node -> Node: Start Timestamp (St)
    Node -> Node: Confirmation Timestamp (Ct)
    activate Node
    Node -> Node: Calculate Latency (L) = Ct - St
    deactivate Node
end
```

- The sequence diagram represents the general flow of the Proof of Importance (PoI) consensus algorithm, where nodes stake tokens, participate in network activity, calculate reputation and importance scores, create and validate blocks, and receive block rewards.
- The "Transaction Latency Calculation" loop is added to the sequence diagram to incorporate the analytical modeling equation for calculating transaction latency in PoI.
- Within the "Transaction Latency Calculation" loop, the Node gathers the necessary

information, which includes the Start Timestamp (St) and Confirmation Timestamp (Ct) of a specific transaction.

- The Node then calculates the transaction latency (L) using the analytical modeling equation: L = Ct - St. This equation subtracts the Start Timestamp from the Confirmation Timestamp to determine the time it takes for a transaction to be confirmed in the PoI mechanism.
- The resulting transaction latency value (L) represents the estimated time it takes for a transaction to be confirmed in the Proof of Importance consensus.
- Please note that the actual implementation of Proof of Importance may vary, and the specific details of reputation, activity metrics, and importance score calculations can differ based on the PoI algorithm used in a particular blockchain network. The sequence diagram provided here is a general representation to demonstrate how the analytical modeling equation for transaction latency can be integrated into the PoI consensus algorithm.

3.3.15. Proof of Authority (PoA)

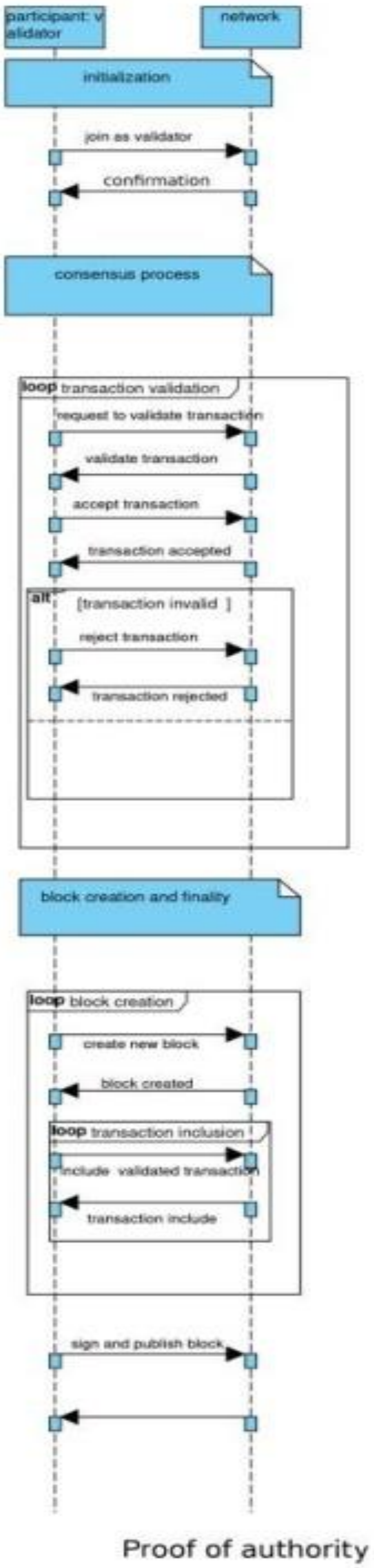


Figure 3.4.5 Proof of Importance

One to one mapping of sequence diagram to analytical modeling of Proof of

Authority

```
participant Validator
participant Network
participant Transaction
participant Block

Note over Validator, Network: Initialization

Validator -> Network: Join as a validator
Network -> Validator: Confirmation

Note over Validator, Network: Consensus process

loop Transaction Validation
    Validator -> Network: Request to validate transaction
    Network -> Validator: Validate transaction
    alt Transaction valid
        Validator -> Network: Accept transaction
        Network -> Validator: Transaction accepted
    else Transaction invalid
        Validator -> Network: Reject transaction
        Network -> Validator: Transaction rejected
    end
end
Note over Validator, Network: Block creation and finality
```

```
loop Block Creation
    Validator -> Network: Create new block
    Network -> Validator: Block created
    loop Transaction Inclusion
        Validator -> Network: Include validated transactions
        Network -> Validator: Transactions included
    end
    Validator -> Network: Sign and publish block
    Network -> Validator: Block added to the blockchain
end

Note over Validator, Network: Calculate PoA Latency

alt PoA Latency Calculation
    Validator -> Network: Gather necessary information (Block Creation Time, Block Confirmation Time)
    Network -> Validator: Information received
    Validator -> Network: Calculate Latency (L = Block Creation Time + Block Confirmation Time)
    Network -> Validator: Latency (L) calculated
else PoA is not applicable
    Validator -> Network: Proceed with regular consensus
end
```

- In the "alt" fragment, the Validator requests the necessary information (Block Creation

Time, Block Confirmation Time) from the Network to calculate the PoA latency.

- The Network provides the information to the Validator.
- The Validator calculates the latency (L) by adding the Block Creation Time and the Block Confirmation Time.
- The Validator sends back the calculated latency (L) to the Network.
- If PoA is not applicable, the Validator proceeds with regular consensus, which is indicated by the "else" fragment.
- Please note that the sequence diagram and the analytical modeling equation provided here are for demonstration purposes only and do not capture the full complexity of a real-world PoA implementation. The actual implementation of PoA may involve additional factors and considerations depending on the specific blockchain network and consensus algorithm used.

3.3.16. Proof of elapsed time (PoET)

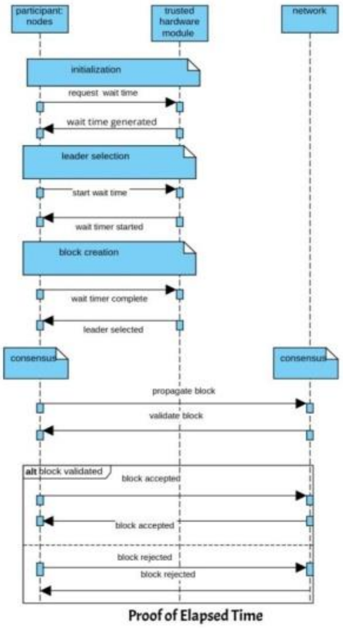


Figure 3.4.6 Proof of elapsed time (PoET)

One to one mapping of sequence diagram to analytical modeling of Proof of

Elapsed Time

```
participant Nodes
participant Trusted Hardware Module (THM)
participant Network
participant Block

Note over Nodes, THM: Initialization

Nodes -> THM: Request wait time
THM -> Nodes: Wait time generated

Note over Nodes, THM: Leader selection

Nodes -> THM: Start wait timer
THM -> Nodes: Wait timer started

Note over Nodes, THM: Block creation

Nodes -> THM: Wait timer complete
THM -> Nodes: Leader selected

Nodes -> Network: Create new block
Network -> Nodes: Block created
```

```
Note over Nodes, Network: Consensus

Nodes -> Network: Propagate block
Network -> Nodes: Validate block
alt Block validated
    Nodes -> Network: Block accepted
    Network -> Nodes: Block accepted
else Block invalid
    Nodes -> Network: Block rejected
    Network -> Nodes: Block rejected
end

Note over Nodes, Network: Block addition

Nodes -> Network: Add block to blockchain
Network -> Nodes: Block added to blockchain

Note over Nodes, THM: Proof of Elapsed Time (PoET) Latency Calculation

alt PoET Latency Calculation
    Nodes -> THM: Gather necessary information (St, Et, Bt, Ct)
    THM -> Nodes: Information received
    opt Calculate Proof Elapsed Time (PET)
        Nodes -> THM: Calculate PET (PET = Et - St)
        THM -> Nodes: PET calculated
    end
    opt Calculate Latency (L)
        Nodes -> THM: Calculate Latency (L = PET * Ct * Bt)
        THM -> Nodes: Latency (L) calculated
    end
else PoET is not applicable
    Nodes -> THM: Proceed with regular consensus
end
```

- In this modified sequence diagram, we added a section to represent the calculation of Proof of Elapsed Time (PoET) latency. This is depicted using the "alt" (alternative) and "opt" (optional) combined fragments in UML sequence diagrams.
- In the "alt" fragment, the Nodes request the necessary information (Start Timestamp, End Timestamp, Block Time, Block Confirmation Threshold) from the Trusted Hardware Module (THM) to calculate PoET latency.
- In the "opt" fragment, the Nodes calculate Proof Elapsed Time (PET) by subtracting the Start Timestamp (St) from the End Timestamp (Et) and Latency (L) by multiplying PET with the Block Confirmation Threshold (Ct) and Block Time (Bt). If PoET is not applicable, the Nodes proceed with regular consensus.

3.3.17. Proof of Burn (PoB)

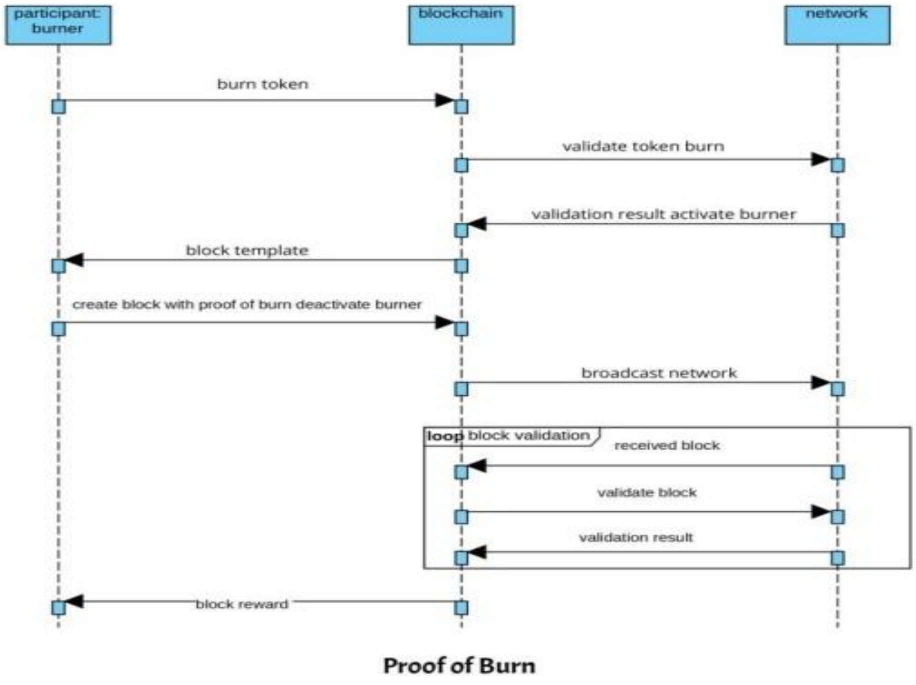


Figure 3.4.7 Proof of Burn (PoB)

One to one mapping of sequence diagram to analytical modeling of Proof of Burn

participant Burner

participant Blockchain

participant Network

Burner -> Blockchain: Burn Tokens

Blockchain -> Network: Validate Token Burn

Network -> Blockchain: Validation Result

activate Burner

Blockchain -> Burner: Block Template

Burner -> Blockchain: Create Block with Proof of Burn

deactivate Burner

Blockchain -> Network: Broadcast Block

```
loop Block Validation
    Network -> Blockchain: Received Block
    Blockchain -> Network: Validate Block
    Network -> Blockchain: Validation Result
end

Blockchain -> Burner: Block Reward

loop Burn Time Calculation
    Burner -> Burner: Gather Necessary Information
    Burner -> Burner: Amount to Burn (B)
    Burner -> Burner: Burning Rate (R)
    activate Burner
    Burner -> Burner: Calculate Burn Time (T) = B / R
    deactivate Burner
end
```

- The sequence diagram represents the general flow of the Proof of Burn mechanism, where the Burner burns tokens, validates the token burn, creates a block with proof of burn, and broadcasts the block to the network.
- The "Burn Time Calculation" loop is added to the sequence diagram to incorporate the analytical modeling equation for calculating the burn time in Proof of Burn.
- Within the "Burn Time Calculation" loop, the Burner gathers the necessary information, which includes the Amount to Burn (B) and the Burning Rate (R) for the

tokens.

- The Burner then calculates the burn time (T) using the analytical modeling equation: T = B / R. This equation divides the Amount to Burn (B) by the Burning Rate (R) to determine the time required to complete the burning process.
- The resulting burn time value (T) represents the estimated time it would take for the participant to complete the burning process based on the burn rate and the desired amount to burn.

3.3.18. Proof of Capacity / Proof of Space (PoC / PoSpace)

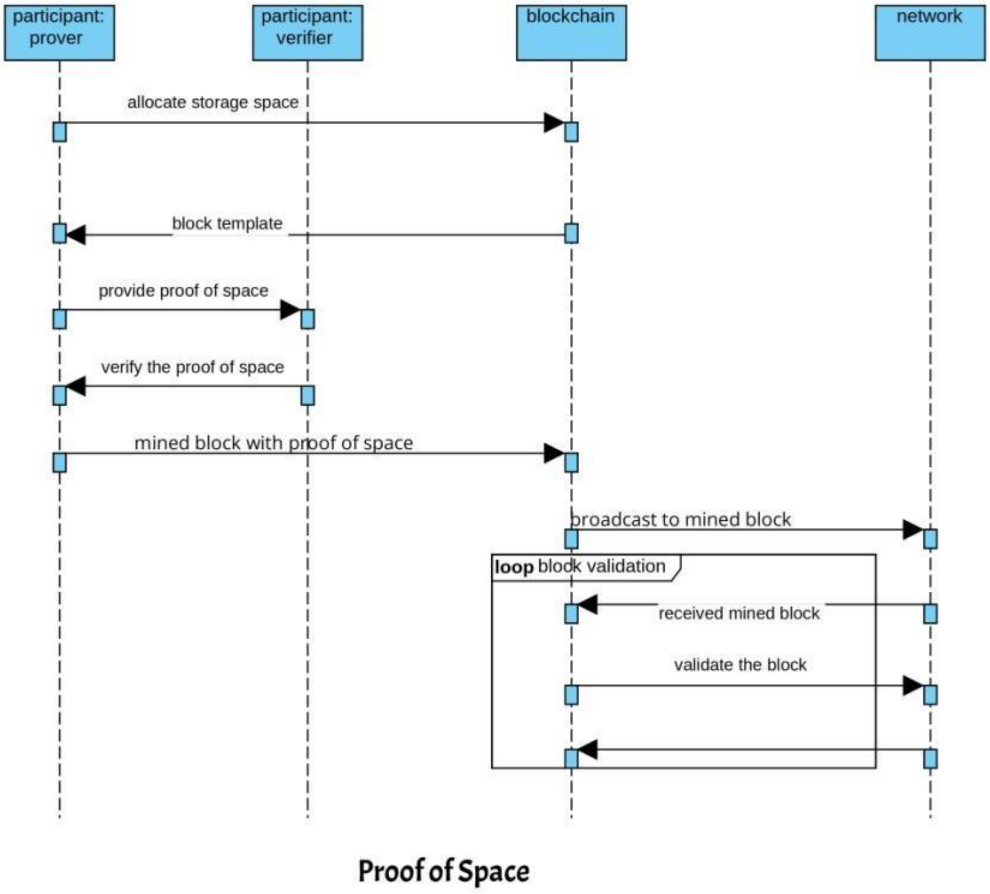


Figure 3.4.8 Proof of Capacity / Proof of Space (PoC / PoSpace)

One to one mapping of sequence diagram to analytical modeling of Proof of Space

```
participant Miner

participant Validator

participant Blockchain

participant Network


loop Space Allocation

   Miner -> Blockchain: Allocate Storage Space

   Blockchain -> Miner: Block Template

end


loop Block Creation

   Miner -> Validator: Provide Proof of Space

   Validator -> Miner: Verify Proof of Space

   alt Proof of Space Validated

      Miner -> Miner: Calculate Latency (L) = Ht

      Miner -> Blockchain: Create Block

   else Proof of Space Invalid

      Miner -> Blockchain: Request New Block Template

   end

end


Blockchain -> Network: Broadcast Mined Block


loop Block Validation

   Network -> Blockchain: Received Block
```

```
    Blockchain -> Network: Validate Block
    Network -> Blockchain: Validation Result
end

Blockchain -> Miner: Block Reward
```

- The sequence diagram represents the general flow of the Proof of Space mechanism, where the Miner allocates storage space and creates blocks by providing Proof of Space to the Validator.
- The "Space Allocation" loop shows the Miner allocating storage space and receiving a block template from the Blockchain.
- In the "Block Creation" loop, the Miner provides Proof of Space to the Validator, who verifies the proof. If the Proof of Space is valid, the Miner calculates the latency (L) using the equation "L = Ht," where Ht is the time taken to perform the required number of hashes.
- If the Proof of Space is validated, the Miner creates a new block and includes the calculated latency (L) in the block.
- If the Proof of Space is invalid, the Miner requests a new block template from the Blockchain.
- The Blockchain then broadcasts the mined block to the Network.
- The "Block Validation" loop shows the Network receiving the block, and the Blockchain validates the block before broadcasting the validation result.
- Finally, the Miner receives the block reward from the Blockchain.

3.3.19. Proof of History (PoH)

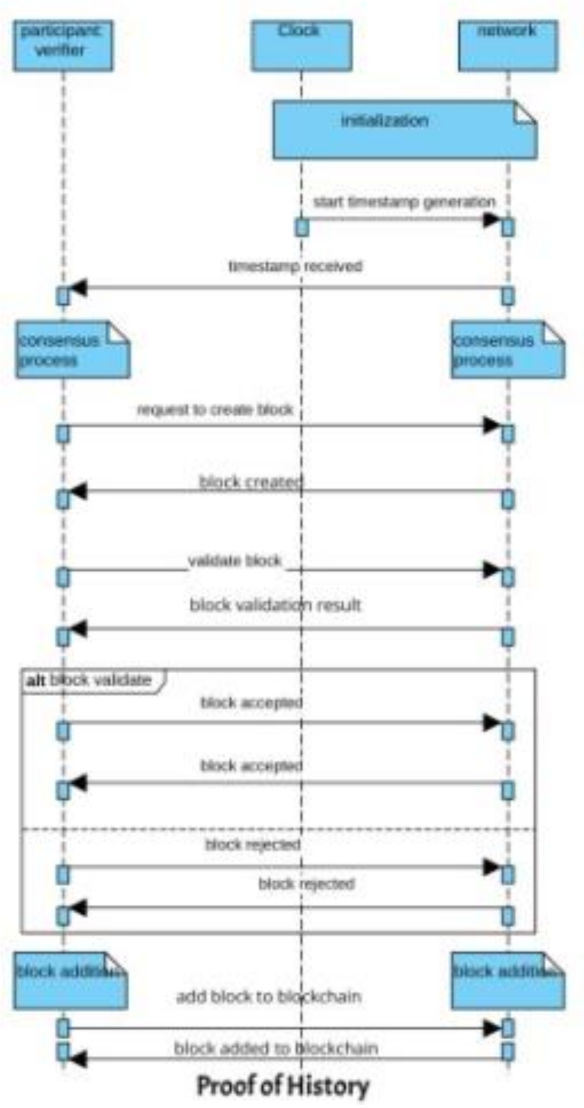


Figure 3.4.9 Proof of History (PoH)

One to one mapping of sequence diagram to analytical modeling of Proof of History

participant Verifier

participant Clock

participant Network

participant Block

```
Note over Clock, Network: Initialization

Clock -> Network: Start timestamp generation
Network -> Verifier: Timestamp received

Note over Verifier, Network: Consensus process

Verifier -> Network: Request to create block
Network -> Verifier: Block created
Verifier -> Network: Validate block
Network -> Verifier: Block validation result
alt Block valid
    Verifier -> Network: Block accepted
    Network -> Verifier: Block accepted
else Block invalid
    Verifier -> Network: Block rejected
    Network -> Verifier: Block rejected
end

Note over Verifier, Network: Block addition

Verifier -> Network: Add block to blockchain
Network -> Verifier: Block added to blockchain

Note over Verifier, Clock: Proof of History Latency Calculation
```

```
alt PoH Latency Calculation
    Verifier -> Clock: Gather necessary information (Start Timestamp, End Timestamp)
    Clock -> Verifier: Information received
    Verifier -> Clock: Calculate Latency (L = End Timestamp - Start Timestamp)
    Clock -> Verifier: Latency (L) calculated
else PoH is not applicable
    Verifier -> Clock: Proceed with regular consensus
end
```

- In the "alt" fragment, the Verifier requests the necessary information (Start Timestamp, End Timestamp) from the Clock to calculate the PoH latency.
- The Clock provides the information to the Verifier.
- The Verifier calculates the latency (L) by subtracting the Start Timestamp from the End Timestamp.
- The Clock sends back the calculated latency (L) to the Verifier.
- If PoH is not applicable, the Verifier proceeds with regular consensus, which is indicated by the "else" fragment.

3.3.20. Proof of Activity

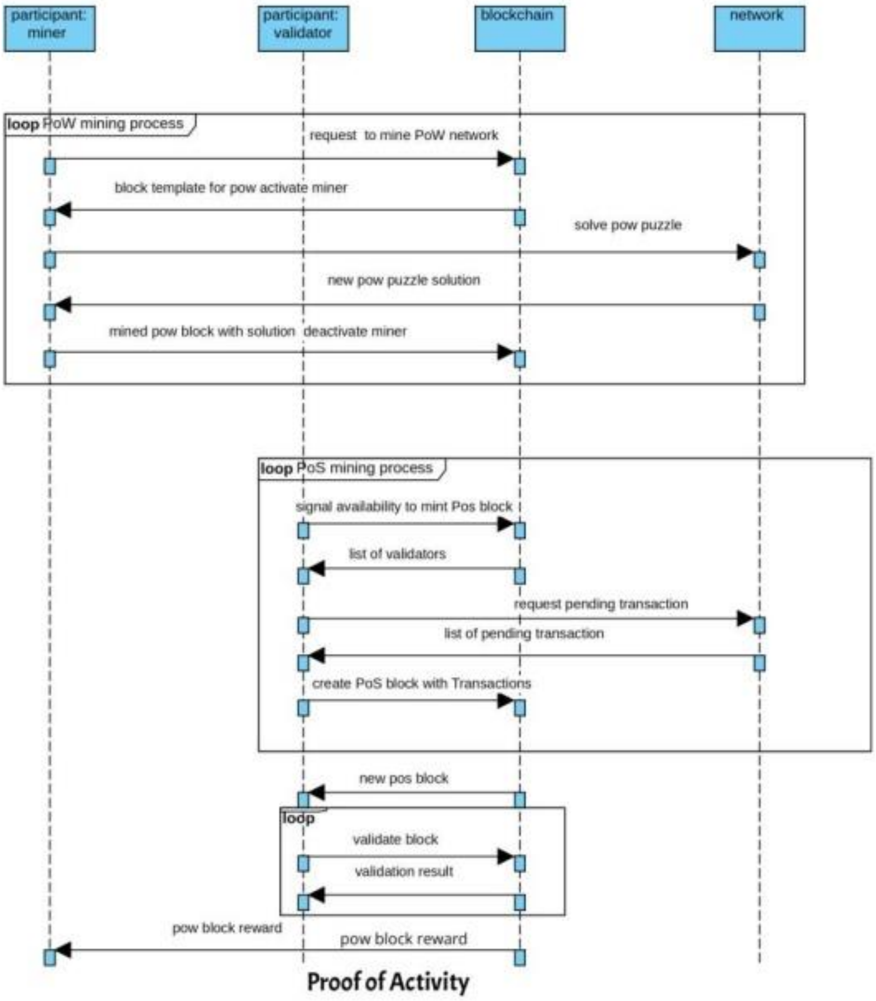


Figure 3.4.10 Proof of Activity

One to one mapping of sequence diagram to analytical modeling of Proof of Activity

participant Miner

participant Validator

participant Blockchain

participant Network

```
loop PoW Mining Process
    Miner -> Blockchain: Request to Mine PoW Block
    Blockchain -> Miner: Block Template for PoW
    activate Miner
    Miner -> Network: Solve PoW Puzzle
    Network -> Miner: New PoW Block Solution
    Miner -> Blockchain: Mined PoW Block with Solution
    deactivate Miner
end

loop PoS Minting Process
    Validator -> Blockchain: Signal Availability to Mint PoS Block
    Blockchain -> Validator: List of Validators
    Validator -> Network: Request Pending Transactions
    Network -> Validator: List of Pending Transactions
    Validator -> Blockchain: Create PoS Block with Transactions
end

Blockchain -> Validator: New PoS Block

loop Block Validation
    Validator -> Blockchain: Validate Block
    Blockchain -> Validator: Validation Result
end
```

Blockchain -> Miner: PoW Block Reward

Blockchain -> Validator: PoS Block Reward

- The sequence diagram represents the hypothetical combination of Proof of Work (PoW) and Proof of Stake (PoS) mechanisms in a Proof of Activity (PoA) concept. The diagram shows the processes of mining PoW blocks and minting PoS blocks.
- In the "PoW Mining Process" loop, the Miner requests to mine a PoW block, solves the PoW puzzle, and creates a new PoW block with a solution.
- In the "PoS Minting Process" loop, the Validator signals availability to mint a PoS block, creates a new PoS block with transactions, and includes it in the blockchain.
- The "Block Validation" loop shows the validation process where the Validator validates the PoS block before broadcasting the validation result.
- Now, to implement the analytical modeling equations for calculating the latencies of PoW and PoS in this context:
- Step 1: Gather the necessary information:
- For PoW:
- Hashrate (H): The computational power of the network in hashes per second for PoW mining.
- Difficulty (D): The difficulty level of the PoW algorithm.
- Block Time (BtPoW): The average time duration between consecutive PoW blocks.
- For PoS:
- Stake (S): The total amount of tokens staked in the PoS component.
- Block Time (BtPoS): The average time duration between consecutive PoS blocks.
- Step 2: Calculate the Proof of Work (PoW) component's latency:
- PoW Latency (PwL): Use the same mathematical steps as described earlier for

calculating the latency of Proof of Work. This involves considering the difficulty, hashrate, and block time specific to the PoW component.

- Step 3: Calculate the Proof of Stake (PoS) component's latency:
- PoS Latency (PsL): Use the same mathematical steps as described earlier for calculating the latency of Proof of Stake. This typically involves considering the stake participation ratio, block time, and the number of validators specific to the PoS component.
- Step 4: Combine the latencies of both components:
- Total Latency (TL): Combine the PoW latency (PwL) and PoS latency (PsL) to get the overall latency of the Proof of Activity mechanism. TL = PwL + PsL

PHASE 4

3.5 Simulation of Consensus Protocol

We simulate many consensus algorithms used in blockchain systems, including Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), and others, in our thorough research. We extend our analytical models for determining latency with real-world data like as hash rates, difficulty levels, staking ratios, voting power, and network sizes to assure accuracy and relevance. We develop these models in NetLogo, a flexible agent-based modelling environment, and simulate the operation of consensus mechanism. We can check how these protocols operate in various network scenarios and evaluate their resilience and scalability thanks to simulations. This study aids in the design and optimization of consensus mechanisms by providing useful insights into the trade-offs between different consensus algorithms.

3.6 Summary

In this chapter, the proposed methodology has been described in detail. The proposed design is segmented into five phases. From exploration of consensus protocol to drive the analytical model of each consensus protocol and map these model to sequence diagram of consensus protocol in block chain then we calculate the latency consensus through simulation on net logo for results validation.

..

CHAPTER 4

Blockchain Based Simulation Model

4.1. Overview

In this chapter, we describe the architecture of blockchain, stakeholders included in our model, introduction to agent based modeling used from simulation of blockchain. Results of simulations are discussed in detail with changing perimeter

4.2. Introduction

Abstract representation of a design of existing system is called model. It describes the dynamics of the system by combining structural, mathematical expression and logical relationship. Whereas, Simulation [34] a quantitative method mimics the behavior of system by executing the model. It is used to predict the “what-if” behavior of system such that how it will respond under certain conditions, introduction of new polices and designs without disturbing the system functioning. Blockchain is build using Agent Based Modeling (ABM) [35] simulation, ABM mimics real life systems having to build formal models. Real life systems have units such as (e.g. animals, institutions, people, cells or atoms) who interact among themselves and with their surroundings. In blockchain simulation we use agents such as nodes, blocks and transactions to interact with each other to model and predict the behavior of system and based on it we can calculate the latency ABM offers accurate prediction by modeling individual units of real world and their actions are explicitly represented in the model.

Existing blockchain simulations are developed with programming languages (such as e.g. java, c++, or python) or special purpose simulation languages but we have opted for NetLogo to develop ABM based blockchain simulation. NetLogo is a modeling environment designed for coding and running agent-based simulations . It can hold large and complex simulation with reasonable speed, run simulations with thousand or more agents, and facilitate interaction with agents at runtime. Hence, NetLogo is well written, easy to learn and easy to install online environment.

4.3. Blockchain Architecture

Blockchain is layered architecture, it can be divided in to three basic layers network layer, consensus layer and incentive layer [37]. We have built our simulation model on these layers such that Network layer creates nodes, controls network level connectivity, and communications between nodes. The consensus layer defines the rules and algorithms used to reach the consensus about state of the blockchain. While incentive layer manages reward distribution after achieving Conesus by participating nodes. For Latency Calculation we will model only two layers of blockchain (i.e. network and consensus layer) incentive layer is out of scope for this thesis

4.3.1. How it works

Blockchain network is a group of devices or connected computers. These devices or connected computers are called nodes. These nodes make up a Blockchain network. All participant nodes in this architecture are connected via Blockchain network, which is responsible for broadcasting a transaction to all nodes in the network. Blockchain Network has no central authority. Transactions generated by nodes are forwarded to the Blockchain network. At the Blockchain network, consensus takes places and the nodes in the network validate each of the transactions. After validation, these transactions are packed in new blocks which is broadcasted to every peer node in the network, and records of these transactions are recorded on their copy of the ledger

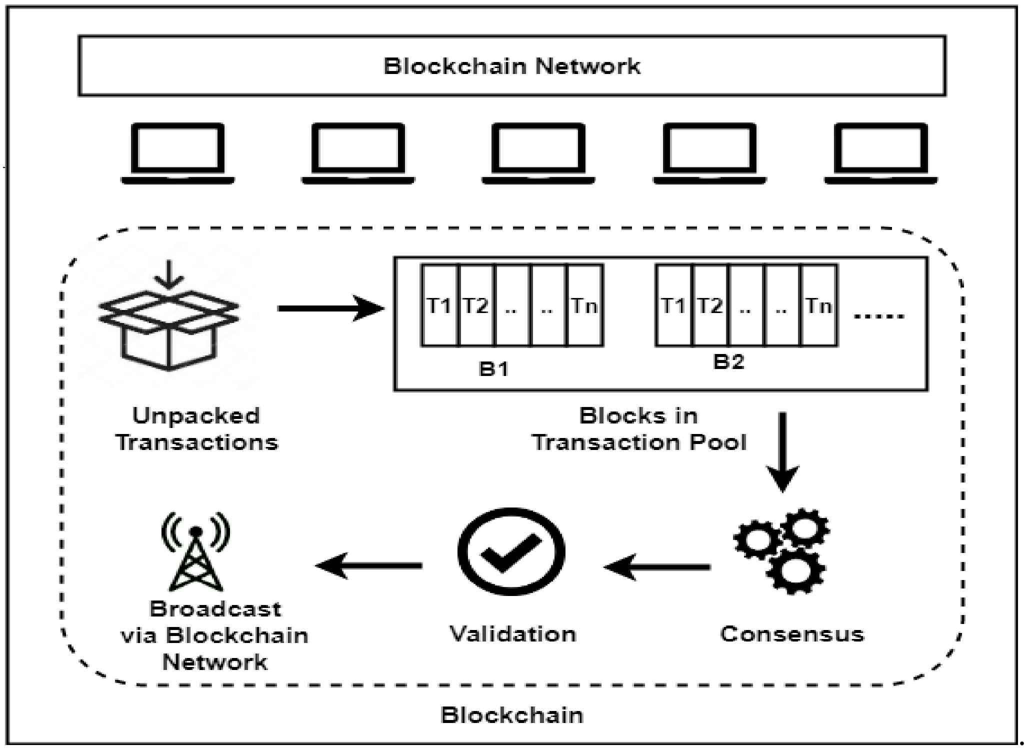


Figure 4.0.1Blockchain network

4.3.1.1. Network Latency

In digital world latency can be defined as the delay encountered between input and output. Here in blockchain we refer it to time required to add next block of transaction .in the blockchain ledger. It is the wait user encounters when he initiate the transaction to its acceptance and addition to chain [38].

4.3.1.2. Layer Architecture

Working of blockchain network is explained in layer architecture model is shown in figure 3.2. Nodes and network protocol setup is in network layer whereas consensus layer have blockchain leger and transaction which is part of node. Validate transaction depends on generate transaction so it can take transaction to validate and add then into block so in-turn validate transaction depends on block for addition of transactions whereas block depends on blockchain ledger for adding newly generated block to ledger. Block also depend on network protocol to broadcast the newly created block to network. Block latency can be calculated from block and network protocol. Incentive layer depends on block from consensus layer to gain block reward. To model this architecture we have used agent based modeling for blockchain simulation and analyze how nodes, blocks, transactions are created, transactions are validated and added into newly created block, then checking the latency of network such that how much time it takes to add new block depending on transaction rate of network.

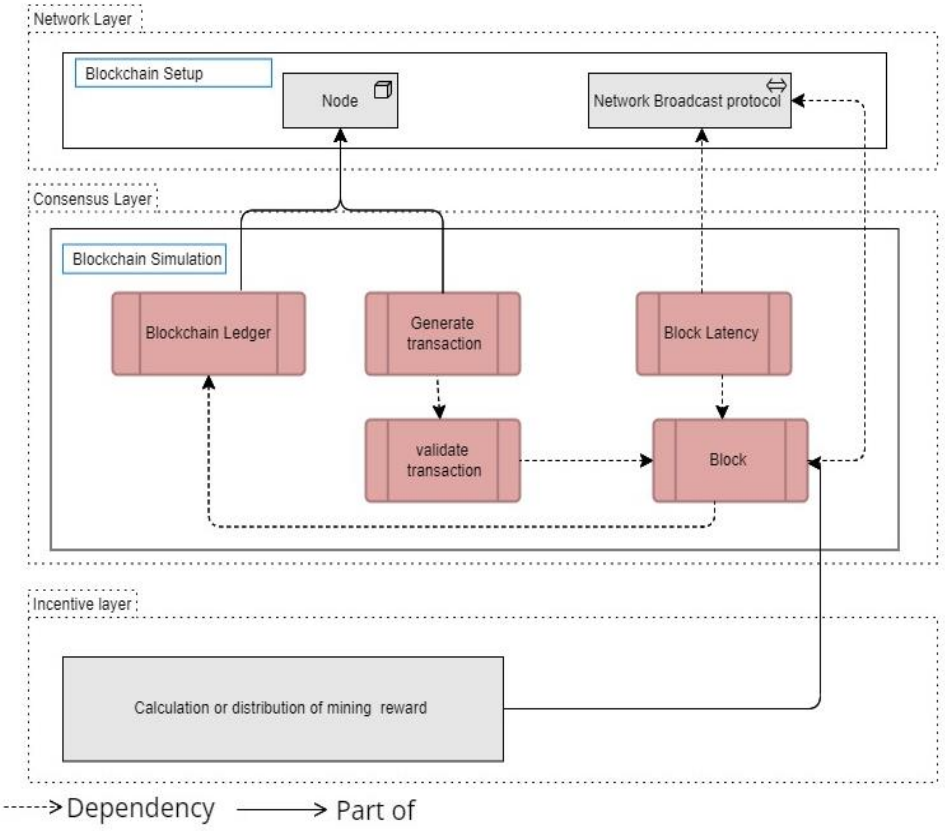


Figure 4.0.1.2Blockchain Layered Architectur

4.3.1.3. Stakeholders in Model

Agent	Responsibilities
Validator Node	Validator node is responsible for validating transaction, adding it to new block and broadcasting this block to network. It can also generate transitions. Maintains blockchain leger.
Non-validator node	Non-validator nodes are responsible for generating transactions like users, then waiting for transaction to be accepted. They also maintain blockchain ledger.
Positive influencer Node	Nodes having positive opinion. Can communicate with network and influence the network positively. These nodes can also be validator or non-validator node. Detail explanation in voter model.
Negative Influencer Node	Nodes having negative opinion. Can communicate with network and influence the network negatively. These nodes can also be validator or non-validator node. Detail explanation in voter model.
Unsure Node	Nodes unsure of their opinion. Can interact with influencers and change its opinion accordingly. These nodes can also be validator or non-validator node. Detail explanation in voter model.

4.3.1.4. Blockchain Simulation Agents

We have three types of agents in model node, block and transaction. Node manages setup of nodes, creating genesis block, generating transactions and setting apart percentage of nodes to be validator from total nodes. Block owns block size, Hash created from date-time-and block who number, validated transaction list, id of blockchain, pre block and next block's hash. If it is not genesis that hash of genesis block. While transaction owns transaction data, transaction owner, waiting time of transaction to be validated, time at which transaction generated and time it got validated in order to calculate latency of network, id of transections blockchain, and some parameters to check if transaction is valid, waiting or added to block. We have some variables which can be varied during simulation to check the effect on results such as number of nodes, transaction rate and percentage of validators in network. By setting these variables to different values every time we can get different simulation results.

4.3.2. Network layer

The blockchain network stores and share history of distributed ledger. So network layer is an important layer while discussing blockchain simulation. Network provides a way of communication for Nodes to achieve consensus and maintain single copy of ledger throughout the network.

This layer defines two process for agent "Node". First is creation of node while the other one is node connectivity. Node creation is responsible for setup of nodes, creating genesis block, generating transactions and setting apart validator nodes. Later one defines the protocol for communication between nodes addressed in proposed model.

4.3.2.1. Simulation

Firstly create Nodes and genesis block then by setting percentage of required validators in network we divide the nodes in network as validator and non- validator node. Number of validators in network can be calculated by$Validators = \ num\ of\ nodes *$ $(\ \%validator * 0.01)$. Figure 3 Flow chart for setup of blockchain network. Such that for have Number of nodes 150, 50% validator nodes we have 75 validator in network of 150 nodes.

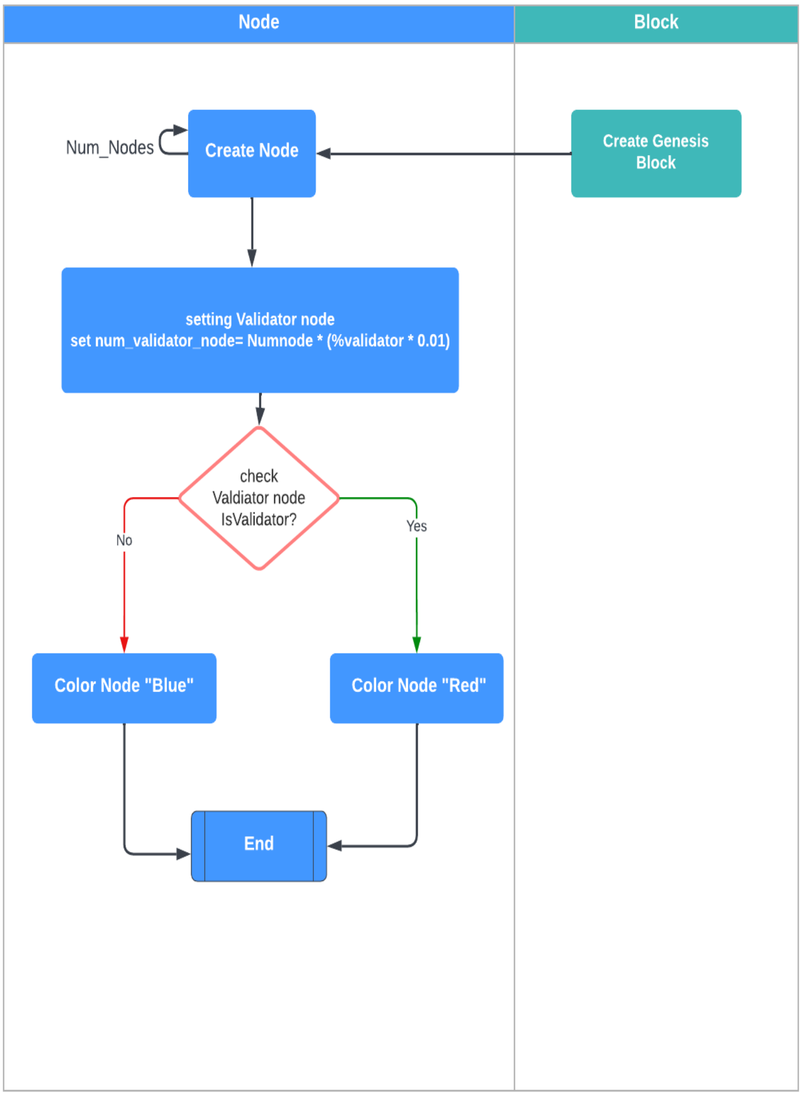
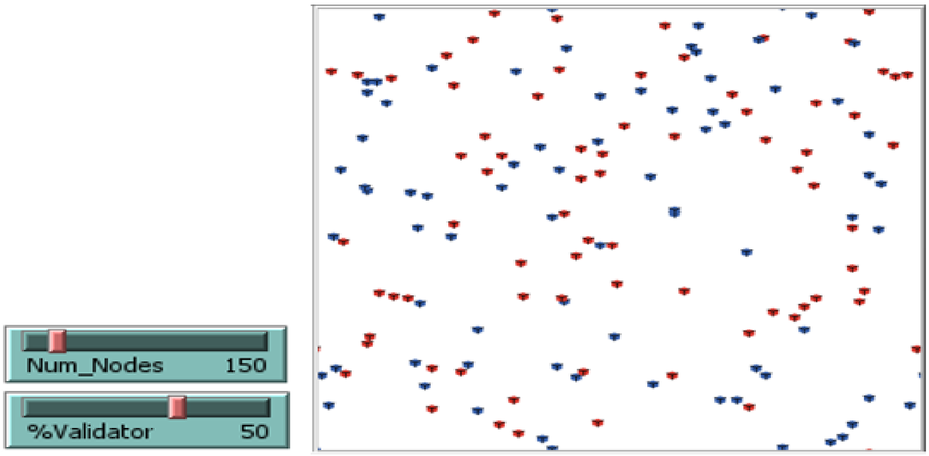


Figure 4.0.2.1 Network setup

4.3.2.2. Results

Suppose we have Number of nodes 150, 50% validator nodes in network then distribution will be as follow figure

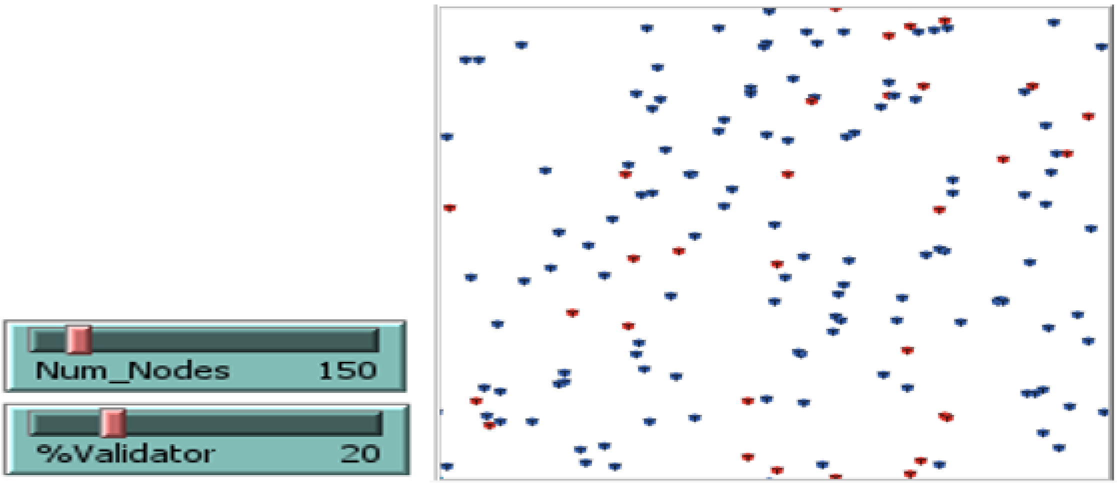


Num_Nodes 150

%Validator 50

Figure 4.3.2.2 Blockchain Setup blue nodes 150, Red %validator nodes 50

We have equal number of validator (Red blocks) and non-validator (Red blocks) nodes. If we set 20% validators then we can see following distribution.



Num_Nodes 150

%Validator 20

Figure 4.3.2.3 Blockchain Setup blue nodes 150, Red %validator nodes 20

Similarly nodes can also be varied with different percentage of validator. This is why we used ABM, it let us create network to our own choice and simulated the environment to see different results every time

4.3.2.4. Consensus Layer

We have already setup the nodes in network layer now it is ready to generate transactions, blocks and create a blockchain. This layer defines algorithms and rules to achieve consensus among nodes of the network. It deals with transaction validation, block generation and verification.

4.3.2.5. Simulation

After creating Nodes and genesis block in network layer now we will see how transactions are generated, processed and packed in a block. As a result we can measure the average delay faced by each transaction before getting blocked and appending to chain. See figure 4.3.2.5 Flow chart for blockchain simulation.

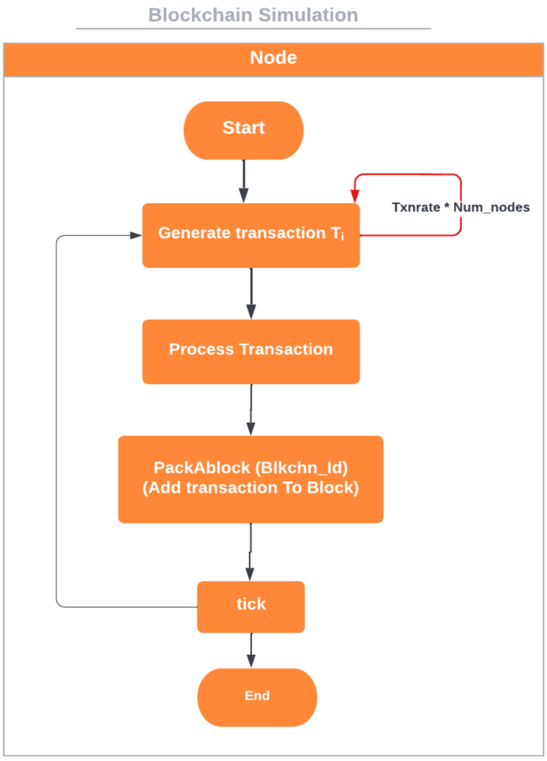


Figure 4.3.2.5 Blockchain simulation

For N number of nodes NetLogo calls randomly each node to run operations iteratively. Tick is built-in procedure which enhances each time procedure runs and ends when user wants to end the function call otherwise can run every time tick increases until unless stopped. As we can see there are 3 procedures called by node to run the blockchain, below is stepwise explanation of each procedure.

a) Generate Transaction

It will generate random transactions based on number of nodes and transaction rate set in network. Such as if we have 150 nodes and transaction rate is 0.4 then according to $Num\_nodes * Trnxrate$ will be 60. So it will run operation random 60 to generate transaction. As we know NetLogo selects any random node from number of nodes to be transection generator and will create one transection and set its data.

b) Process Transaction

Check for transections awaiting to be validated. Setting the random delay for each transaction and calculating sum of average delay of each transaction which is packed in block.

c) PackAblock

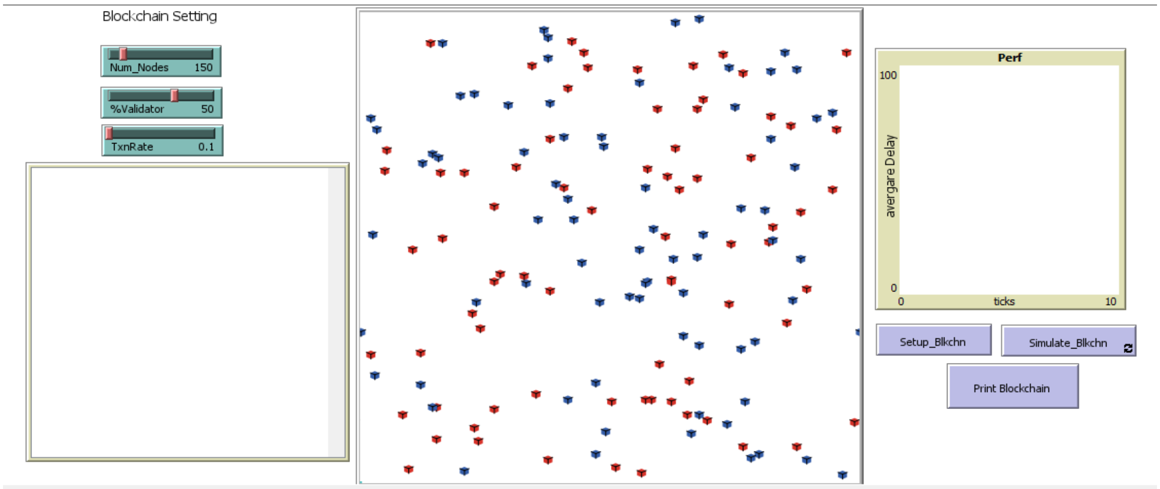
If the number of validated transactions are ready to be packed then make ordered list of transaction according to pre-defined number of transaction that can be added. Keep the count of added transactions. Create block hash by concatenating date and time with id. Once the transactions are packed validator call all nodes to copy the packed transections and the hash. Update the ordered list to all nodes, add new block to existing blockchain ledger of all nodes.

Validator nodes start working on next block by using previous block’s hash to add on chain. Which shows that they have accepted the block.

All nodes trust the longest chain for adding block. It follows the Bitcoins longest chain rule. After each addition of block to chain we have measured the delay encountered by block and plotted the measure of the average delay encountered by each transaction from generation to getting blocked. In subsequent section we will discuss in detail, how changing transaction rate, number of nodes and percentage of validators effects the throughput of network.

4.3.2.6. Results

We will start with setting up the blockchain interface with 3 sliders namely number of nodes, %validator (number of validators in network), Transaction rate can be varied from 0.1 to 1. (Generating transaction rate per sec). then 3 buttons for setting up blockchain as explained in network layer, blockchain simulation forever button which runs the blockchain to add block at every tick until press again to end to process, print blockchain prints the state of blockchain ledger across all nodes. For interface see figure 4.3.2.6


Blockchain Setting

Num_Nodes 150

%Validator 50

TxnRate 0.1

Figure 4.3.2.6 Blockchain Simulation Interface

We will run the simulation for 100 ticks, num_nodes 150, 50% validators and transaction rate 0.1. It will generate random transactions based on number of nodes and transaction rate set in network such that$Num_nodes * Trnxrate$. For average delay measures there is auto scale plot having tick on x-axis and average delay on y-axis. Calculation of average delay is as follow

$$\begin{aligned} &\mathit{Sum\ of\ average\ delay} \\ &= \mathit{delay\ time\ of\ previous\ transaction} \\ &+ (\mathit{end\ time\ of\ current\ transaction} - \mathit{start\ time\ of\ current\ transaction}) \end{aligned}$$

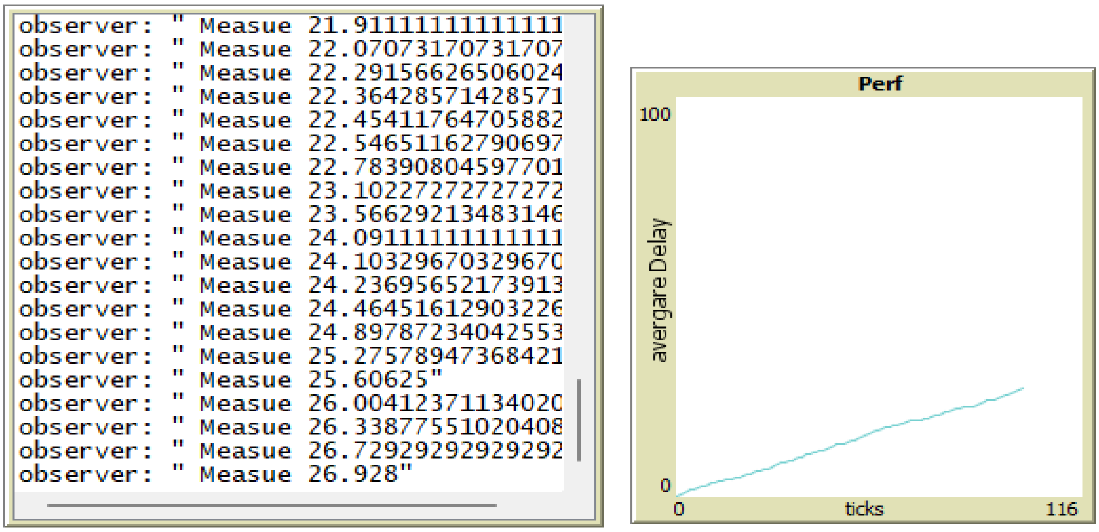
$\mathit{Count\ transactions} = \mathit{counter\ of\ packed\ transaction.}$ It enhances every time transaction is added to block.

$$\mathit{Average\ delay} = (\mathit{sum\ of\ average\ delay\ /\ count\ transaction})$$

These equations runs 5 times at each tick if number of transaction to be packed in block are 5.

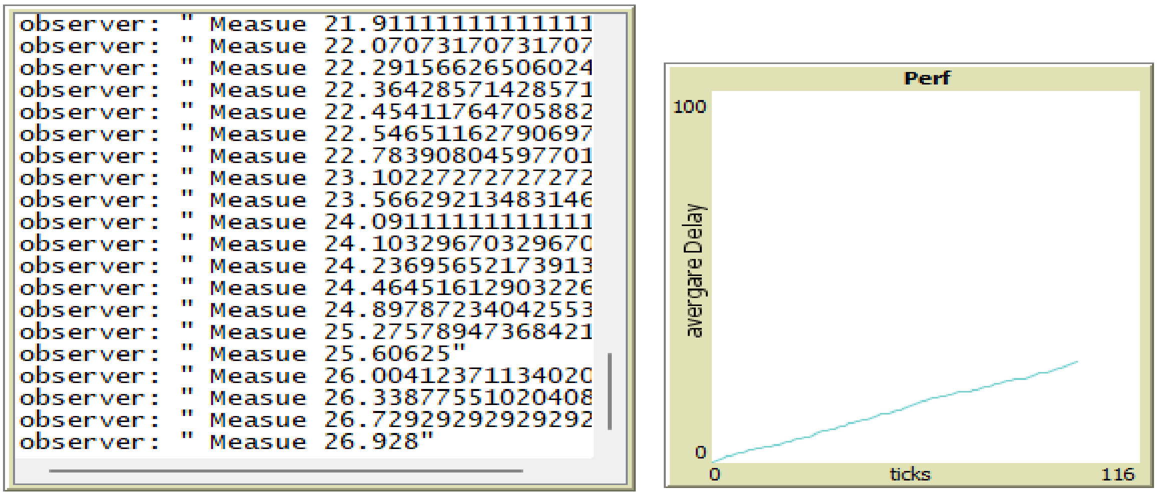
4.3.2.7. Changing transaction rate

To analyze the effect of change in transaction rate on network latency, we will run the simulation for constant 100 ticks, number of nodes 150, 50% validators and varying transaction rate 0.1 and then for 1.



```
observer: " Measue 21.91111111111111
observer: " Measue 22.07073170731707
observer: " Measue 22.29156626506024
observer: " Measue 22.36428571428571
observer: " Measue 22.45411764705882
observer: " Measue 22.54651162790697
observer: " Measue 22.78390804597701
observer: " Measue 23.10227272727272
observer: " Measue 23.56629213483146
observer: " Measue 24.09111111111111
observer: " Measue 24.10329670329670
observer: " Measue 24.23695652173913
observer: " Measue 24.46451612903226
observer: " Measue 24.89787234042553
observer: " Measue 25.27578947368421
observer: " Measue 25.60625"
observer: " Measue 26.00412371134020
observer: " Measue 26.33877551020408
observer: " Measue 26.72929292929292
observer: " Measue 26.928"
```

Figure 4.0.2.7 Average Delay for Transaction rate 0.1



```
observer: " Measue 21.91111111111111
observer: " Measue 22.07073170731707
observer: " Measue 22.29156626506024
observer: " Measue 22.36428571428571
observer: " Measue 22.45411764705882
observer: " Measue 22.54651162790697
observer: " Measue 22.78390804597701
observer: " Measue 23.10227272727272
observer: " Measue 23.56629213483146
observer: " Measue 24.09111111111111
observer: " Measue 24.10329670329670
observer: " Measue 24.23695652173913
observer: " Measue 24.46451612903226
observer: " Measue 24.89787234042553
observer: " Measue 25.27578947368421
observer: " Measue 25.60625"
observer: " Measue 26.00412371134020
observer: " Measue 26.33877551020408
observer: " Measue 26.72929292929292
observer: " Measue 26.928"
```

Figure 4.3.2.8Average delay for Transaction Rate 1

Table 4.3.2.1 Nodes 150, %validator 50%, transaction Rate variable 0.1-1.0

Transaction Rate	Average Delay
0.1	11.103
0.2	22.07
0.6	24.31
1	26.92

Keeping other parameters constant, change transaction rate 0.1 to 1 we can see from plot and table 2 that average delay is increased such that 0.1 = 11.103 and 1.0 = 26.92. Hence, we can say $Transaction\, rate \propto Average\; Delay$ such that transaction rate is directly proportional to average delay more transaction rate more average delay to add the block, less transaction rate less time to add a block also more transactions waiting to get validated because now validator are busy in validating the transactions.

4.3.2.8. Changing %validator

To analyze, how changing percentage of validators in network effects the latency? We will keep constant 150 nodes in network and calculate average delay for 100 ticks. The plots and table have been used to display the results. Such that table list the results of plot in sequence. To make it more understandable through numbers.

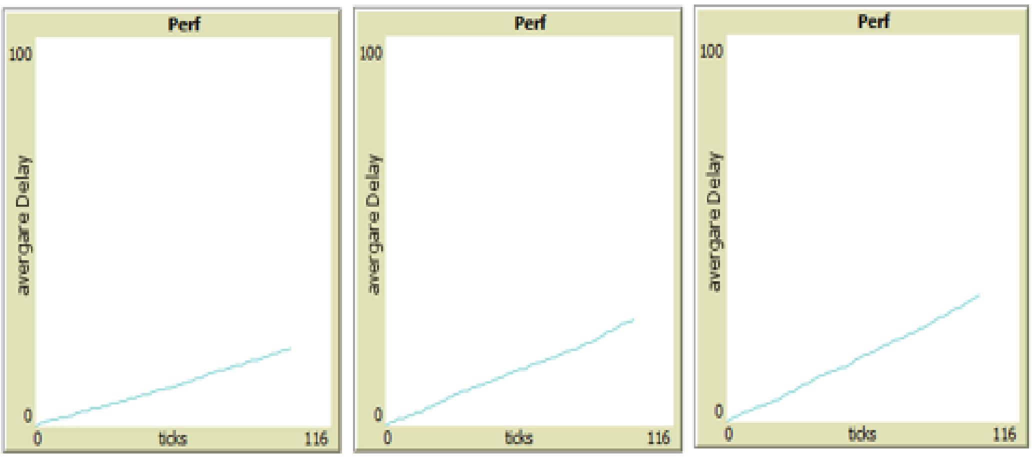


Figure 4.3.2.8.1 Average delay from left to right transaction rate 0.2, 0.7, and 1.

Table 4.3.2.8.1 Node 150, %validator 30% transaction rate 0.2, 0.7, and 1.

%Validator	**Transaction Rate**	**Average Delay**
30%	0.2	19.91
30%	0.7	26.96
30%	1	32.45

Changing validator percentage to 50% to observe the difference in average delay.

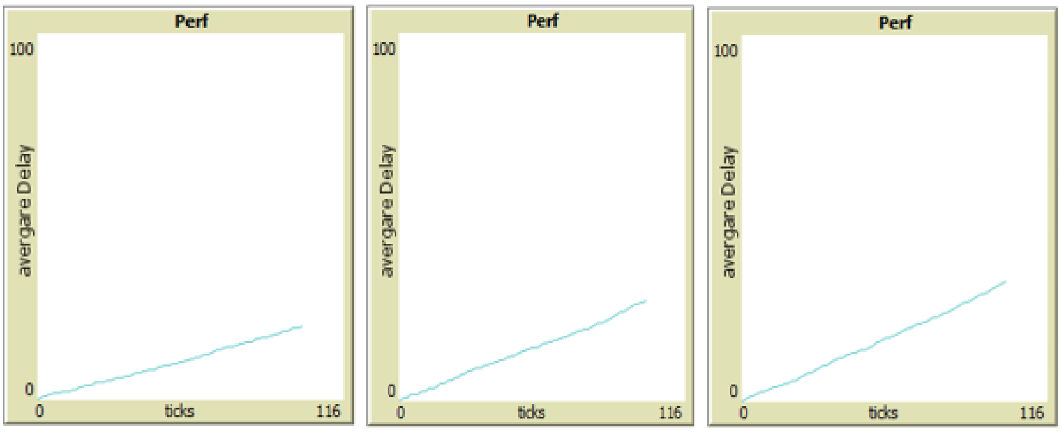


Figure 4.3.2.8.2Average delay from left to right transaction rate 0.2, 0.7, and 1

Table 4.3.2.8.2 Node 150, %validator 50% transaction rate 0.2, 0.7, and 1.

%Validator	Transaction Rate	Average Delay
50%	0.2	22.07
50%	0.7	25.54
50%	1	26.92

Plot shows little rise in average delay (blue line) linearly with every tick. According to table 3 and 4, for specific percentage of validator when we increase transaction rate as a result average delay to add block to chain increases. Hence, we can say that $\%validator \propto \frac{1}{Average\ Delay}$ such that more number of validators, less average delay whereas less validators more average delay and more transactions waiting to be validating which adds to latency of network.

4.3.2.9. Changing Number of nodes

To analyze, how changing percentage number of nodes in network effects the latency? We will keep 40% validator nodes in network, transaction rate 0.3 constant and calculate average delay for 100 ticks. The plots and table have been used to display the results. Such that table list the results of plot in sequence to make it more understandable.

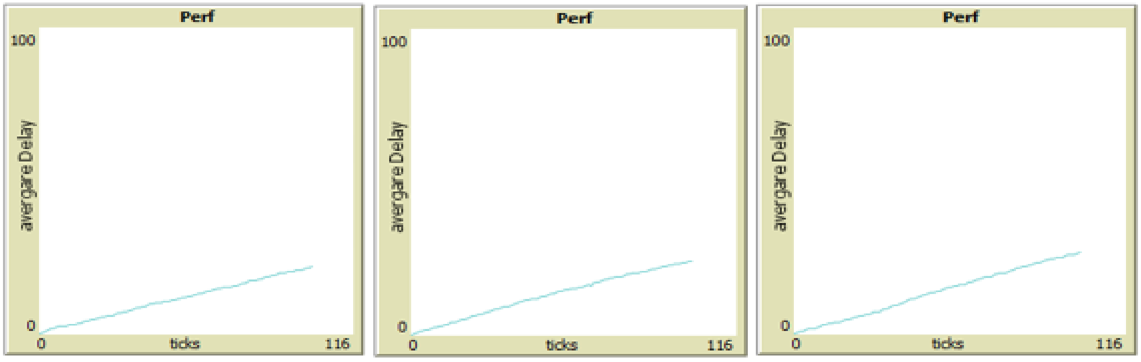


Figure 4.3.2.9.1 Average delay from left to right for varying number of Nodes 150, 300, and 600 in network.

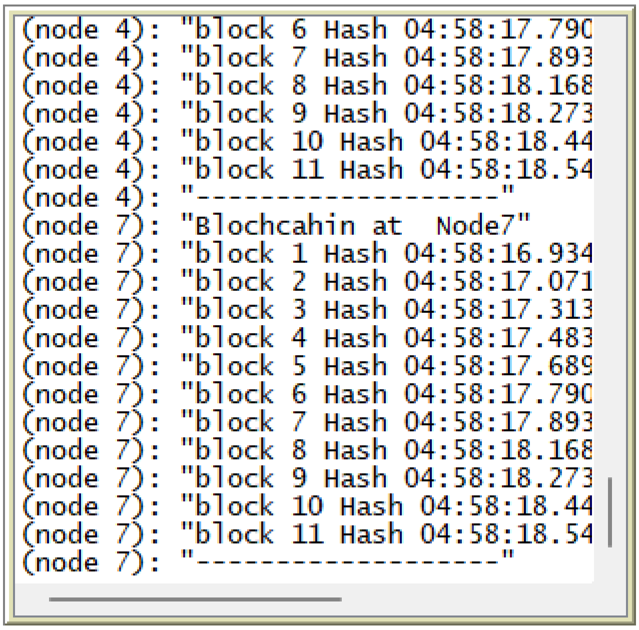
Table 4.3.2.9.1 Average delay for varying number of Nodes in network.

Nodes	**Average Delay**
150	21.76
300	24.35
600	26.57

Plot shows little rise in average delay (blue line) linearly with every tick. According to table 5, for small number of nodes latency is less whereas for increasing number of nodes latency of blockchain increases. Also time to generate and add block of transactions also increase which in turn give boast to network latency. Hence, we can say if we increase transaction rate then $Number\ of\ nodes\ \propto Average\ Delay$ such that more number of nodes more average delay vice versa. We have to set transaction rate low in such a huge network e.g. network having nodes 1000 and transaction rate 0.1 it will still result into increased average delay with more transactions still waiting to be validated, delay in acceptance of transaction and block addition to chain. More the network gets bigger more latency problems introduced for larger network transaction rate should be as lower as possible to milliseconds. So size of network plays vital role in latency of network.

4.3.2.10. Print Blockchain

We know when block is validated it is broadcasted so everyone on network can update its ledger accordingly to maintain the immutability and trust. We have implemented this procedure which shows the blockchain leger maintained by all nodes in network. It has blockhash of each block on the chain but hides the transactions.



```
(node 4): "block 6 Hash 04:58:17.790
(node 4): "block 7 Hash 04:58:17.893
(node 4): "block 8 Hash 04:58:18.168
(node 4): "block 9 Hash 04:58:18.273
(node 4): "block 10 Hash 04:58:18.44
(node 4): "block 11 Hash 04:58:18.54
(node 4): "-------------------"
(node 7): "Blochcahin at  Node7"
(node 7): "block 1 Hash 04:58:16.934
(node 7): "block 2 Hash 04:58:17.071
(node 7): "block 3 Hash 04:58:17.313
(node 7): "block 4 Hash 04:58:17.483
(node 7): "block 5 Hash 04:58:17.689
(node 7): "block 6 Hash 04:58:17.790
(node 7): "block 7 Hash 04:58:17.893
(node 7): "block 8 Hash 04:58:18.168
(node 7): "block 9 Hash 04:58:18.273
(node 7): "block 10 Hash 04:58:18.44
(node 7): "block 11 Hash 04:58:18.54
(node 7): "-------------------"
```

Figure 4.3.2.10.1 Blockchain Leger at each node in network

4.4. Conclusion

This chapter includes the introduction to Agent Based Modeling, blockchain architecture, simulation, stakeholders in our model, and results of simulation. Our research models the implantation and interaction of network and consensus layer. At network level it setup the blockchain environment, while on consensus layer it performs the simulation of blockchain to generate transactions, add block to the blockchain and plot the latency of network. Incentive layer is out of scope of this research but can be implemented in future work.

Latency of blockchain can be described as delay encountered by network during processing of transaction to addition of block to blockchain. By blockchain simulation AB modeling we have experimented with varying different parameters in network such as number of nodes, transaction rate and percentage of validators to see the effect on latency of network. Results has shown the pattern such that for small blockchains, latency is less similarly for bigger blockchain, latency is more because transaction rate is low, number of nodes are more to generate and validate the transaction. Transaction rate and network size has direct proportionality with network latency whereas percentage of validator is inversely proportional to latency because if we have more validator then latency will be low vice versa. Hence, if we want low latency than we need to have balanced number of validator not less it will effect latency but not too many they will gain power in network and could hack the network , low transaction rate for big or small blockchain. Lower latency can shape the blockchain to move forward.

CHAPTER 5

Simulation of Consensus Protocol in Blockchain for Latency Calculation

5.1. Overview

In this chapter, we added a consensus layer in our existing model for check the latency delay in validation we use two protocol Pos and Pow. Our focus is on calculating latency between consensus node selection and average delay among nodes, enhancing system efficiency. This ensures a secure and responsive network.

5.2. Introduction

In the last chapter, we explored how blocks are broadcasted and the associated delays when a new block is added to the blockchain ledger. Back then, we didn't have a specific consensus layer for validation. However, things are different now. We've introduced a consensus layer because we're diving deep into the critical analysis of the consensus protocol, particularly concerning latency. Our focus is on enhancing validation through the addition of Proof of Stake (PoS) for node validation. and also check with adding proof of work After this validation process, we're keen on examining the latency delays. Specifically, we're looking at the time it takes between selecting a high-stake nodewhile work with proof of stake and measuring delays under varying transaction rates both high and low. It's a detailed analysis to ensure our system performs optimally in different scenarios.

5.3. Blockchain POS Simulation

We have already described the architecture of blockchain in chapter 4. Now we will implement and simulated pos and calculate latency

5.3.1. Blockchain POS Simulation Agents

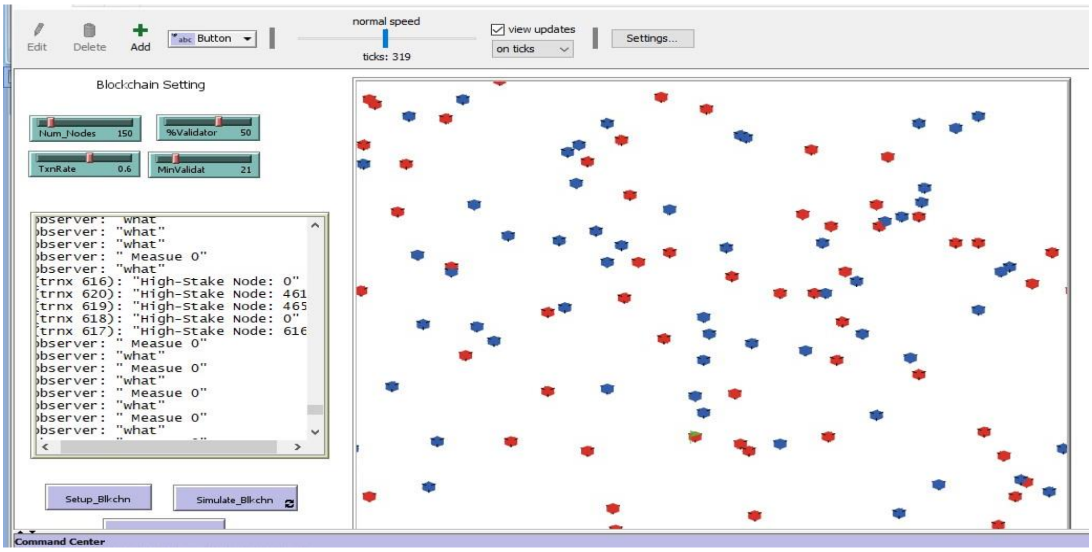
We have three types of agents in model node, block and transaction. Node manages setup of nodes, creating genesis block, generating transactions and setting apart percentage of nodes to be validator from total nodes. Block owns block size, Hash created from date-time-and block who number, validated transaction list, id of blockchain, pre block and next block's hash. If it is not genesis that hash of genesis block. While transaction owns transaction data, transaction owner, waiting time of transaction to be validated, time at which transaction generated and time it got validated in order to calculate latency of network, id of transactions blockchain, and some parameters to check if transaction is valid, waiting or added to block. We have some variables which can be varied during simulation to check the effect By setting these variables to different values every time we can get different simulation results to analyze latency of network.

5.3.2. Network Layer

Network layer is same as define in chapter 4, this layer defines two process for agent “Node”. First is creation of node while the other one is node connectivity. Node creation is responsible for setup of nodes, creating genesis block, generating transactions and setting apart validator nodes. Later one defines the protocol for communication between nodes addressed in proposed model.

5.3.3. Consensus Layer

Our network nodes are all set up, ready to generate transactions, blocks, and create a blockchain. This layer sets the rules and algorithms for nodes to agree on things, called consensus. Now, we've added a Proof of Stake (PoS) consensus layer, dealing with validating transactions, creating blocks, and making sure they're legit.In this PoS layer, a high-stake node is chosen as the validator to give a nod to transactions. Once that's done, the process kicks off. We've also introduced a nifty module for calculating delays after each validation, ensuring smooth blockchain addition. Sometimes, delays happen when spreading the word about a new block among nodes. Simultaneously, other nodes might create a new block without knowing about the freshly shared one, creating a fork where nodes see different versions of the blockchain. This is where our consensus layer steps in, resolving these conflicts. Our research focus is on figuring out the delays that pop up during the validation process of the consensus and then cross-validating our design model to make sure our latency calculations are on point.



Num_Nodes 150
%Validator 50
TxnRate 0.6
MinValidat 21

```
bserver:  what
bserver: "what"
bserver: "what"
bserver: " Measue 0"
bserver: "what"
(trnx 616): "High-Stake Node: 0"
(trnx 620): "High-Stake Node: 461
(trnx 619): "High-Stake Node: 465
(trnx 618): "High-Stake Node: 0"
(trnx 617): "High-Stake Node: 616
bserver: " Measue 0"
bserver: "what"
bserver: " Measue 0"
bserver: "what"
bserver: " Measue 0"
bserver: "what"
bserver: " Measue 0"
bserver: "what"
```

Figure 5.3.3 proof of stake simulation

5.3.3.1. Simulation

Transactions are generated, processed and packed in a block. As a result we can measure the average delay faced by each transaction before getting blocked and appending to chain

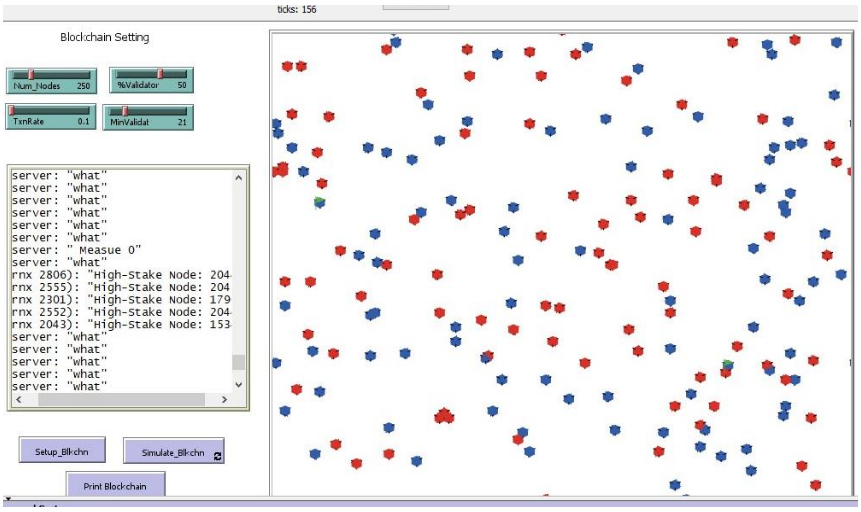
5.3.3.2. Results

To analyze the effect of latency using added the pos consensus layer we change parameter and traction rate and observe what affect occur on high transaction rate and low transaction rate

.

Number of nodes	Validations	Transaction rate	Selection of high stake node	Trxn
250%	50	0.1	Node 2044	trnx 2806
250%	50	0.1	Node 2048	trnx 2555
250%	50	0.1	Node: 1790	trnx 2301
250%	50	0.1	Node: 2044	trnx 2552
250%	50	0.1	Node: 1534	trnx 2043

Table 5.3.3.2 selection of high stake node



```
server: "what"
server: "what"
server: "what"
server: "what"
server: "what"
server: "what"
server: " Measue 0"
server: "what"
rnx 2806): "High-Stake Node: 204
rnx 2555): "High-Stake Node: 204
rnx 2301): "High-Stake Node: 179
rnx 2552): "High-Stake Node: 204
rnx 2043): "High-Stake Node: 153
server: "what"
server: "what"
server: "what"
server: "what"
server: "what"
```

Figure 5.3.3.2 Selection of high Stake Node

5.3.3.3. Latency delay after each Validation

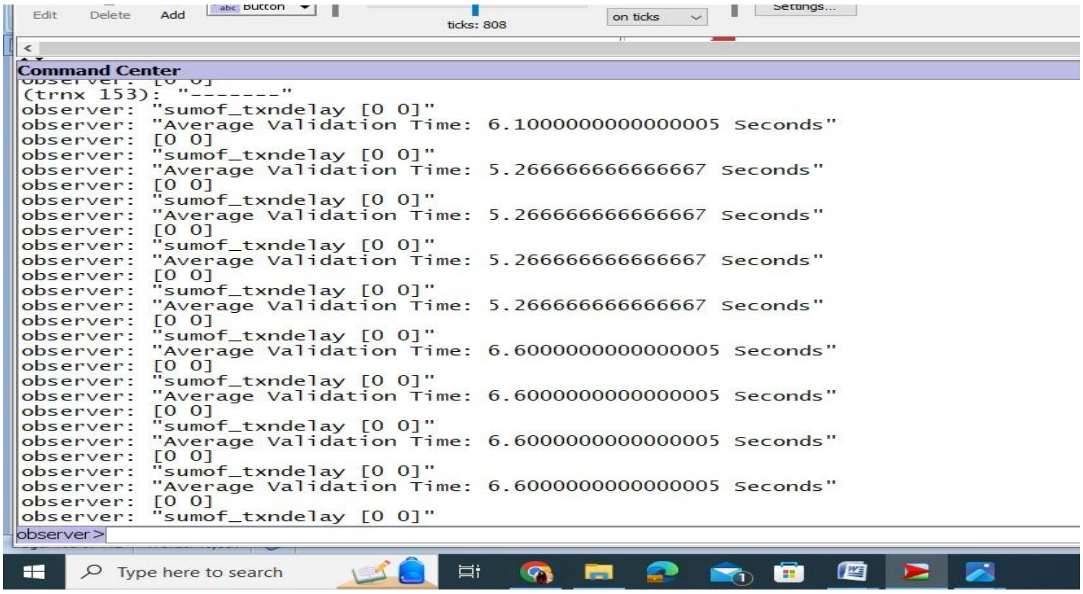
Table 5.3.3.3 Latency Delay after Validation

Transaction number	Sum of Delay	Average validation latency delay
(trnx 308): "-------"	sumof_txndelay [0 0]	40.6 Seconds
(trnx 309): "-------"	sumof_txndelay [0 0]	37.375 Seconds
(trnx 310): "-------"	sumof_txndelay [0 0]	47.3 Seconds
(trnx 771): "-------"	sumof_txndelay [0 0]	845.8999999999999 Seconds
(trnx 158): "……."	sumof_txndelay [0 0]	3.3 Seconds

As we have low transaction rate delay is minimum as we have high transaction rate the delay time increase

5.3.3.4. Low Transaction rate

As we have the low number of transaction rate the delay is minimum due to less number of transaction rate the latency delay rate is less

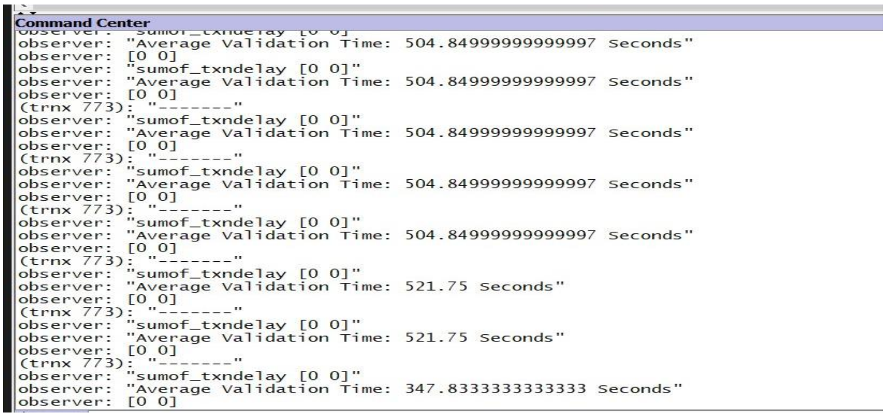


```
observer: [0 0]
(trnx 153): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.1000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
```

Figure 5.3.3.4 low transaction rate

5.3.3.4.1. High Transaction rate

As we have high number of transaction rate delay is maximum cause high number of transaction rate validation take lot of time to validate so time increase and delay occurmost



Command Center

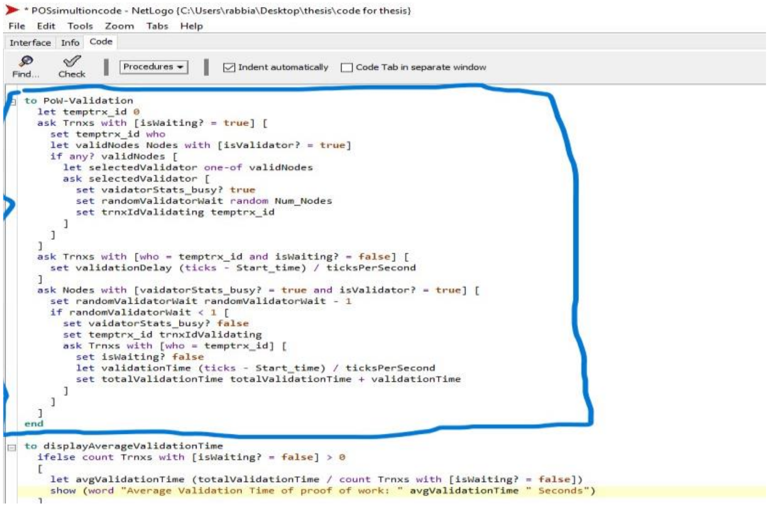
```
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 521.75 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 521.75 seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 347.8333333333333 seconds"
observer: [0 0]
```

Figure 5.3.3.4.1 High Transaction rate

5.4. Blockchain POW Simulation

We use same environment as we mentioned above we add pow conesnus layer in oru existing model now check validation delay

5.4.1.Result



* POSsimultioncode - NetLogo {C:\Users\rabbia\Desktop\thesis\code for thesis}
File Edit Tools Zoom Tabs Help
Interface Info Code
Find... Check
Procedures

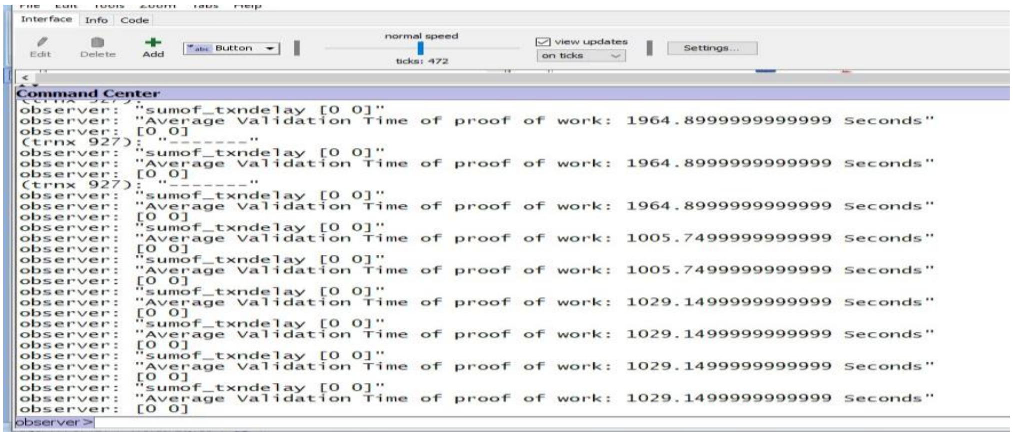
☑ Indent automatically □ Code Tab in separate window

```
to PoW-Validation
  let temptrx_id 0
  ask Trnxs with [isWaiting? = true] [
    set temptrx_id who
    let validNodes Nodes with [isValidator? = true]
    if any? validNodes [
      let selectedValidator one-of validNodes
      ask selectedValidator [
        set vaidatorStats_busy? true
        set randomValidatorWait random Num_Nodes
        set trnxIdValidating temptrx_id
      ]
    ]
  ]
  ask Trnxs with [who = temptrx_id and isWaiting? = false] [
    set validationDelay (ticks - Start_time) / ticksPerSecond
  ]
  ask Nodes with [vaidatorStats_busy? = true and isValidator? = true] [
    set randomValidatorWait randomValidatorWait - 1
    if randomValidatorWait < 1 [
      set vaidatorStats_busy? false
      set temptrx_id trnxIdValidating
      ask Trnxs with [who = temptrx_id] [
        set isWaiting? false
        let validationTime (ticks - Start_time) / ticksPerSecond
        set totalValidationTime totalValidationTime + validationTime
      ]
    ]
  ]
end

to displayAverageValidationTime
  ifelse count Trnxs with [isWaiting? = false] > 0
  [
    let avgValidationTime (totalValidationTime / count Trnxs with [isWaiting? = false])
    show (word "Average Validation Time of proof of work: " avgValidationTime " Seconds")
  ]
```

Figure 5.4.1 PoW validation

5.4.2.High transaction rate

As we have high number of transaction rate delay is maximum cause high number of transaction rate validation take lot of time to validate so time increase and delay occurmost

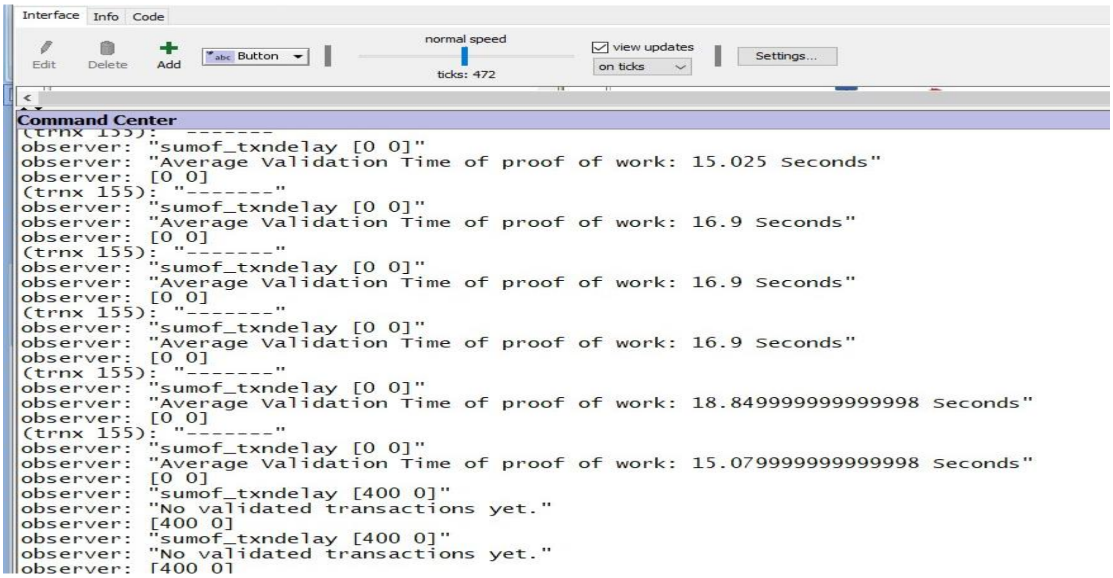


```
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1964.8999999999999 seconds"
observer:  [0 0]
(trnx 927):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1964.8999999999999 seconds"
observer:  [0 0]
(trnx 927):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1964.8999999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1005.7499999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1005.7499999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1029.1499999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1029.1499999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1029.1499999999999 seconds"
observer:  [0 0]
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 1029.1499999999999 seconds"
observer:  [0 0]
```

Figure 5.4.2 high transaction

5.4.3.Low Transaction Rate

As we have the low number of transaction rate the delay is minimum due to less number of transaction rate the latency delay rate is less



```
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 15.025 Seconds"
observer:  [0 0]
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 16.9 Seconds"
observer:  [0 0]
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 16.9 Seconds"
observer:  [0 0]
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 16.9 Seconds"
observer:  [0 0]
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 18.849999999999998 Seconds"
observer:  [0 0]
(trnx 155):  "-------"
observer:  "sumof_txndelay [0 0]"
observer:  "Average Validation Time of proof of work: 15.079999999999998 Seconds"
observer:  [0 0]
observer:  "sumof_txndelay [400 0]"
observer:  "No validated transactions yet."
observer:  [400 0]
observer:  "sumof_txndelay [400 0]"
observer:  "No validated transactions yet."
observer:  [400 0]
```

Figure 5.4.3 low transaction rate

5.5. Conclusion

In this chapter, we conducted a detailed examination of the delay in validating transactions using the POS and POW consensus protocol. We also closely observed how the latency is influenced by the quantity of transactions, both high and low. Our simulations revealed that as the number of transactions increases, the impact on latency becomes more significant, leading to longer validation times due to a large number of consecutive transactions. Conversely, when the transaction rate is low, the delay is minimized, resulting in a more efficient validation process.

CHAPTER 6

Cross Validation of Analytics Model and Simulation Results

6.1. Overview

In this chapter, we validate the latency delay results by comparing the output of our proof-of-stake (PoS) simulation with the analytical model specifically designed for latency calculation in our existing blockchain model. This comparison ensures the accuracy and reliability of our simulated latency measurements against the theoretical expectations.

6.2. Results of POS Simulation Model

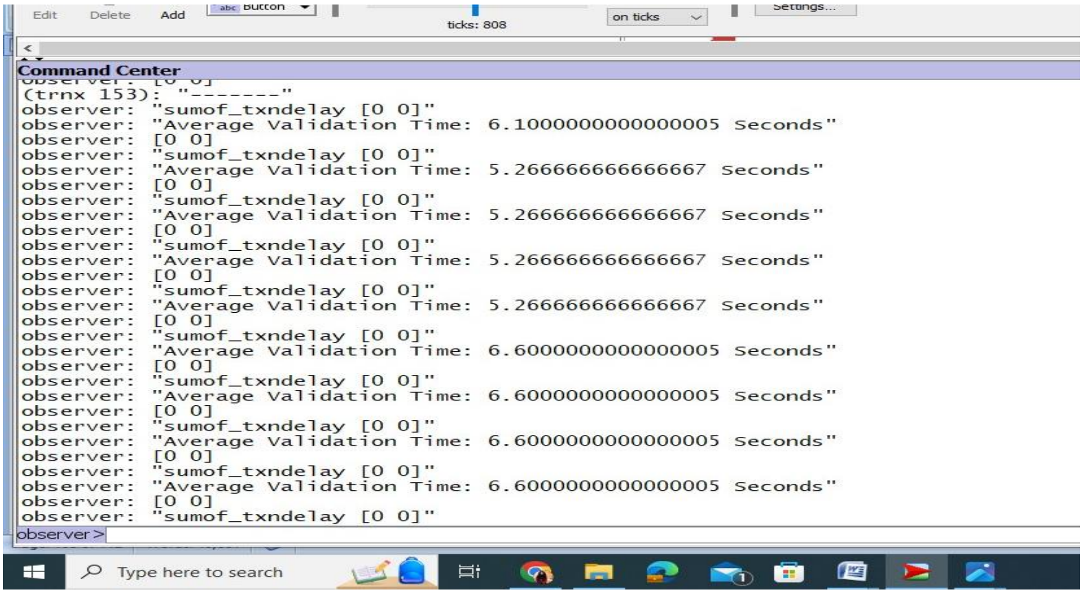
Table 6.1 Result for POS simulation model

Transaction number	**Sum of Delay**	**Average validation latency delay**
(trnx 308): "-------"	sumof_txndelay [0 0]	40.6 Seconds
(trnx 309): "-------"	sumof_txndelay [0 0]	37.375 Seconds
(trnx 310): "-------"	sumof_txndelay [0 0]	47.3 Seconds
(trnx 771): "-------"	sumof_txndelay [0 0]	845.8999999999999 Seconds
(trnx 158): "……."	sumof_txndelay [0 0]	3.3 Seconds

As we have low transaction rate delay is minimum as we have high transaction rate the delay time increase

6.2.1.Low Transaction rate

As we have the low number of transaction rate the delay is minimum due to less number of transaction rate the latency delay rate is less

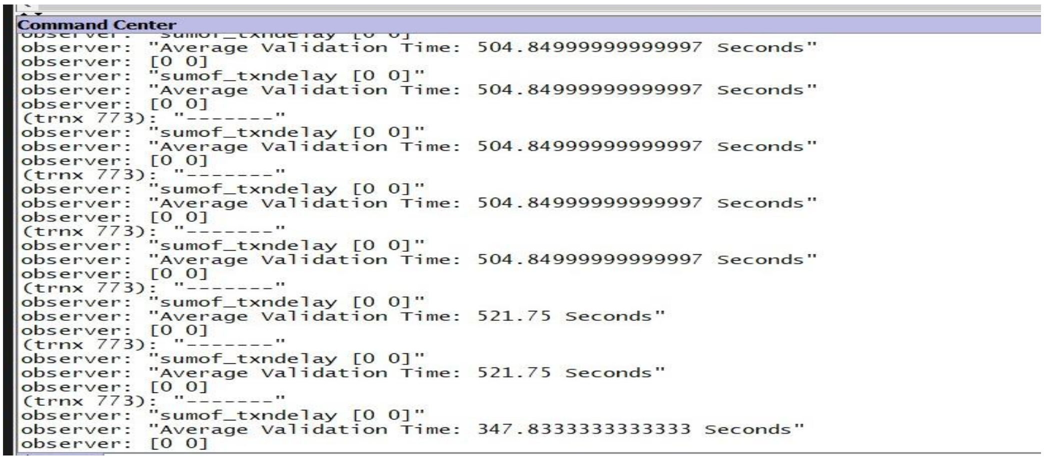


```
observer: [0 0]
(trnx 153): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.1000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 5.266666666666667 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 6.6000000000000005 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer>
```

Figure 6.1.1 Pos low transaction rate

6.2.2.High Transaction rate

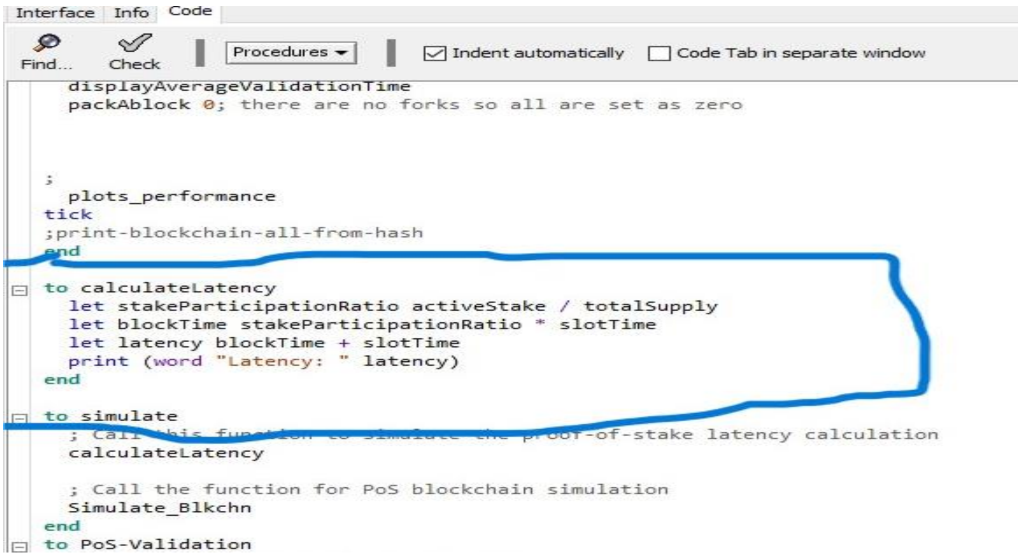
As we have high number of transaction rate delay is maximum cause high number of transaction rate validation take lot of time to validate so time increase and delay occurmost.



```
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 Seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 504.84999999999997 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 521.75 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 521.75 Seconds"
observer: [0 0]
(trnx 773): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 347.8333333333333 Seconds"
observer: [0 0]
```

Figure 6.1.2 Pos High Transaction rate

6.3. Results by using analytical models for latency calculation



☑ Indent automatically □ Code Tab in separate window

```
  displayAverageValidationTime
  packAblock 0; there are no forks so all are set as zero



;
  plots_performance
tick
;print-blockchain-all-from-hash
end

to calculateLatency
  let stakeParticipationRatio activeStake / totalSupply
  let blockTime stakeParticipationRatio * slotTime
  let latency blockTime + slotTime
  print (word "Latency: " latency)
end

to simulate
  ; Call this function to simulate the proof-of-stake latency calculation
  calculateLatency

  ; Call the function for PoS blockchain simulation
  Simulate_Blkchn
end
to PoS-Validation
```

Figure 6.3 Analytical models for pos latency delay

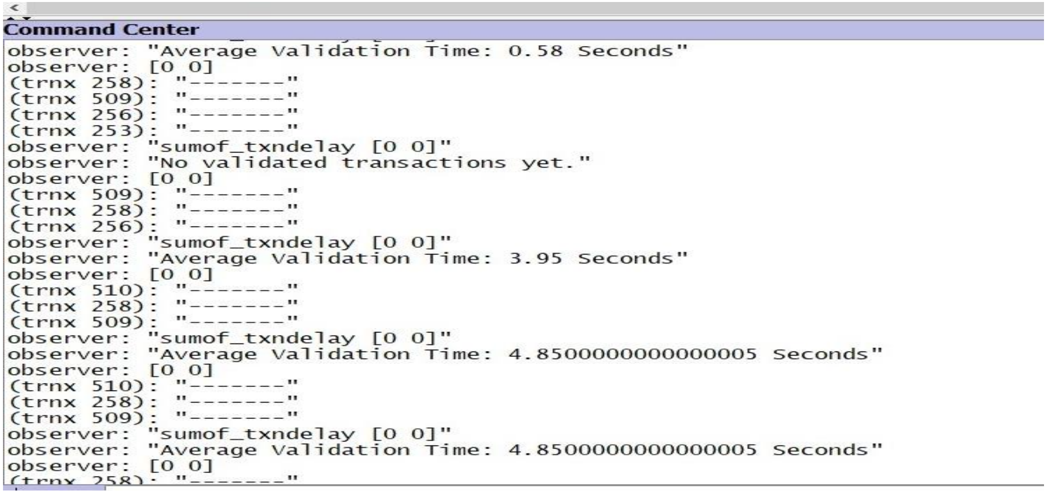
6.4. Results of Analytical Model for Latency Calculation

Table 6.4 Results of pos Analytical model

Transaction number	Sum of Delay	Average validation latency delay
trnx 258 "……"	sumof_txndelay [0 0]	4.8500000000000005 Seconds
trnx 512: "-------"	sumof_txndelay [0 0]	3.5250000000000004 Seconds
trnx 3839:"….."	sumof_txndelay [0 0]	120.66666666666669 Seconds
trnx 6900 "….."	sumof_txndelay [0 0]	1007.1999999999998 Seconds
trxn 256 "……"	sumof_txndelay [0 0]	3.95 Seconds

6.4.1.Low Transaction rate

As we have the low number of transaction rate the delay is minimum due to less number of transaction rate the latency delay rate is less and this thing we prove in other

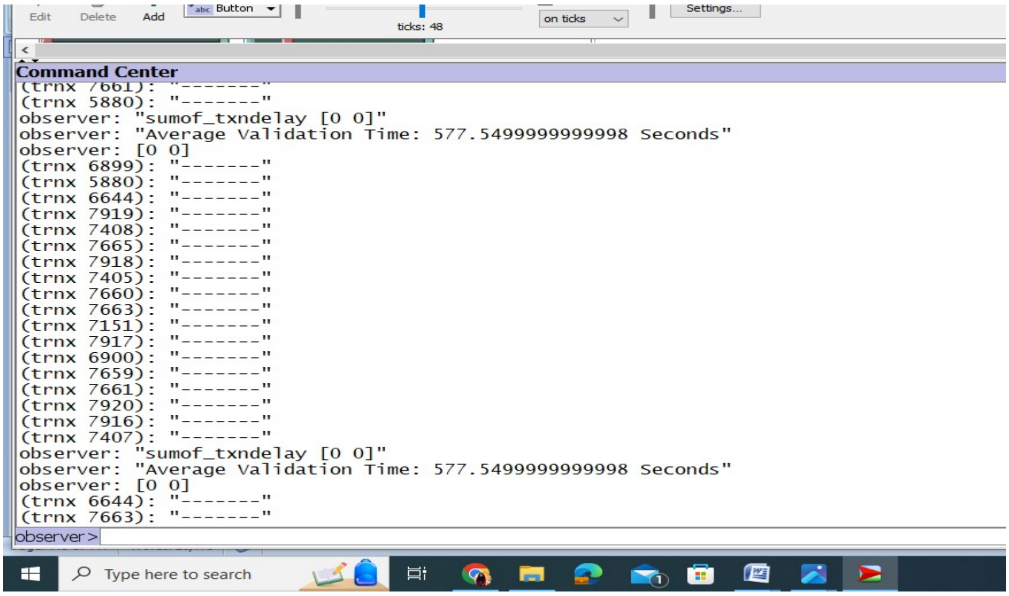


```
Command Center
observer: "Average Validation Time: 0.58 Seconds"
observer: [0 0]
(trnx 258): "-------"
(trnx 509): "-------"
(trnx 256): "-------"
(trnx 253): "-------"
observer: "sumof_txndelay [0 0]"
observer: "No validated transactions yet."
observer: [0 0]
(trnx 509): "-------"
(trnx 258): "-------"
(trnx 256): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 3.95 Seconds"
observer: [0 0]
(trnx 510): "-------"
(trnx 258): "-------"
(trnx 509): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 4.8500000000000005 Seconds"
observer: [0 0]
(trnx 510): "-------"
(trnx 258): "-------"
(trnx 509): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 4.8500000000000005 Seconds"
observer: [0 0]
(trnx 258): "-------"
```

Figure 6.3.1 Pos Analytical model Low Transaction Rate

6.4.2.High Transaction rate

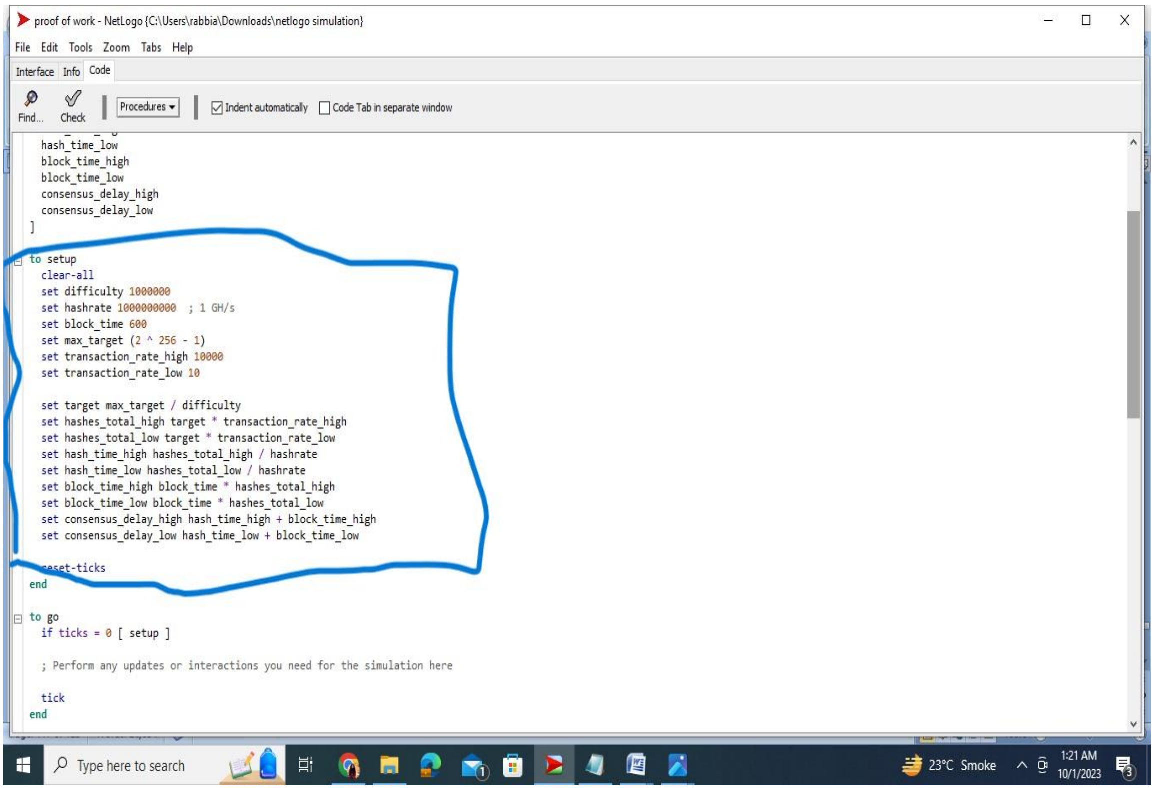
As we have high number of transaction rate delay is maximum cause high number of transaction rate validation take lot of time to validate so time increase and delay occur most.



```
Command Center
(trnx 7661): "-------"
(trnx 5880): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 577.5499999999998 Seconds"
observer: [0 0]
(trnx 6899): "-------"
(trnx 5880): "-------"
(trnx 6644): "-------"
(trnx 7919): "-------"
(trnx 7408): "-------"
(trnx 7665): "-------"
(trnx 7918): "-------"
(trnx 7405): "-------"
(trnx 7660): "-------"
(trnx 7663): "-------"
(trnx 7151): "-------"
(trnx 7917): "-------"
(trnx 6900): "-------"
(trnx 7659): "-------"
(trnx 7661): "-------"
(trnx 7920): "-------"
(trnx 7916): "-------"
(trnx 7407): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 577.5499999999998 Seconds"
observer: [0 0]
(trnx 6644): "-------"
(trnx 7663): "-------"
observer >
```

Figure 6.3.2 Pos analytical model High Transaction Rate

6.5. Results by using PoW analytical models for latency calculation

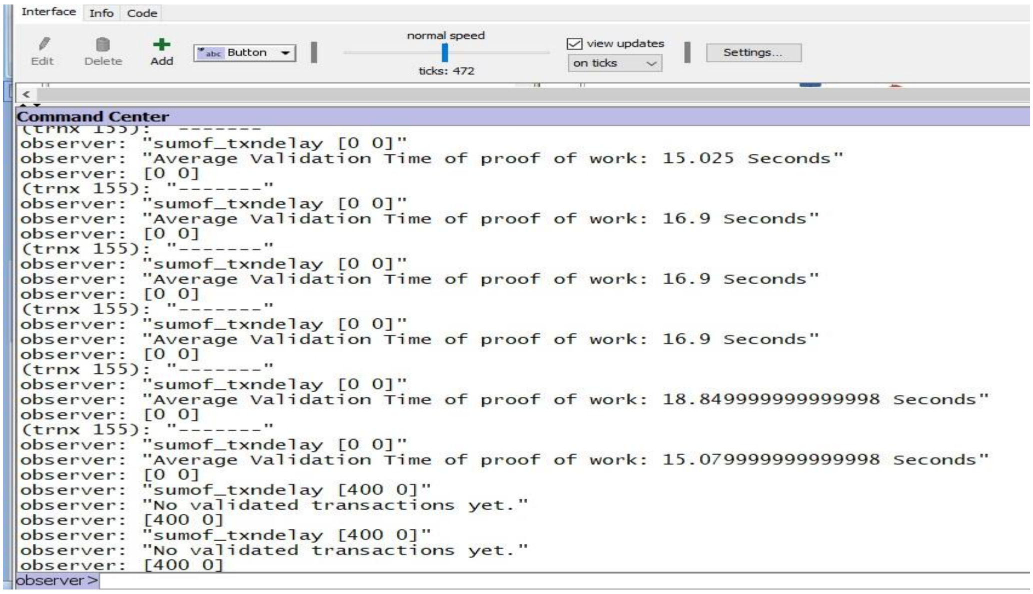


proof of work - NetLogo {C:\Users\rabbia\Downloads\netlogo simulation}
File Edit Tools Zoom Tabs Help
Interface Info Code
Find... Check Procedures ▾ ☑ Indent automatically ☐ Code Tab in separate window
Type here to search
23°C Smoke
1:21 AM
10/1/2023

```
  hash_time_low
  block_time_high
  block_time_low
  consensus_delay_high
  consensus_delay_low
]

to setup
  clear-all
  set difficulty 1000000
  set hashrate 1000000000  ; 1 GH/s
  set block_time 600
  set max_target (2 ^ 256 - 1)
  set transaction_rate_high 10000
  set transaction_rate_low 10

  set target max_target / difficulty
  set hashes_total_high target * transaction_rate_high
  set hashes_total_low target * transaction_rate_low
  set hash_time_high hashes_total_high / hashrate
  set hash_time_low hashes_total_low / hashrate
  set block_time_high block_time * hashes_total_high
  set block_time_low block_time * hashes_total_low
  set consensus_delay_high hash_time_high + block_time_high
  set consensus_delay_low hash_time_low + block_time_low

  reset-ticks
end

to go
  if ticks = 0 [ setup ]

  ; Perform any updates or interactions you need for the simulation here

  tick
end
```

Figure 6.5 analytical model pow added in code

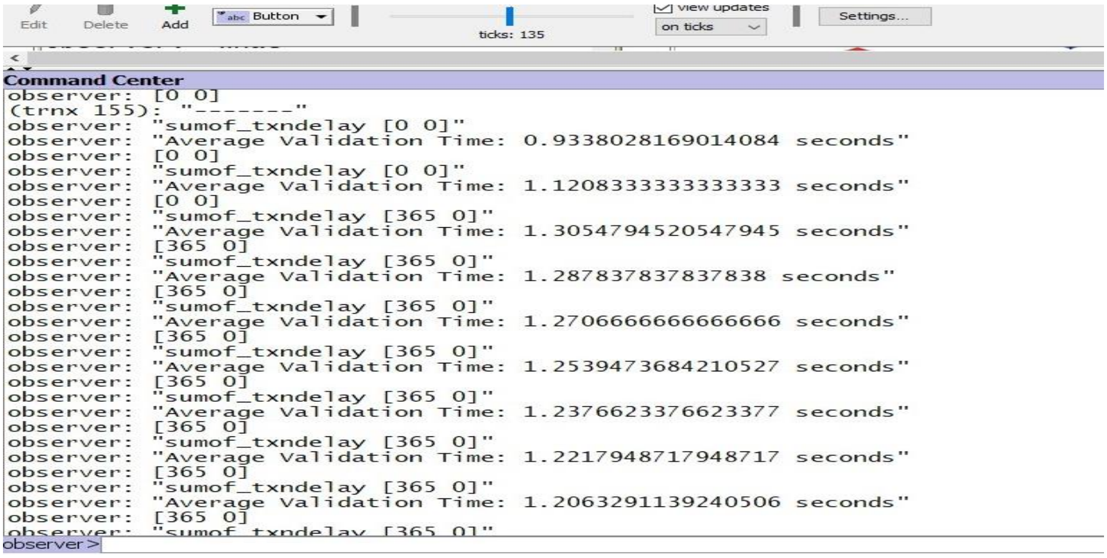
6.5.1.High trasnction rate



```
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 15.025 Seconds"
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 16.9 Seconds"
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 16.9 Seconds"
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 16.9 Seconds"
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 18.849999999999998 Seconds"
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time of proof of work: 15.079999999999998 Seconds"
observer: [0 0]
observer: "sumof_txndelay [400 0]"
observer: "No validated transactions yet."
observer: [400 0]
observer: "sumof_txndelay [400 0]"
observer: "No validated transactions yet."
observer: [400 0]
observer >
```

Figure 6.5.1 Pow high transaction rate

6.5.2.Low transaction rate



```
observer: [0 0]
(trnx 155): "-------"
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 0.9338028169014084 seconds"
observer: [0 0]
observer: "sumof_txndelay [0 0]"
observer: "Average Validation Time: 1.1208333333333333 seconds"
observer: [0 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.3054794520547945 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.287837837837838 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.2706666666666666 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.2539473684210527 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.2376623376623377 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.2217948717948717 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer: "Average Validation Time: 1.2063291139240506 seconds"
observer: [365 0]
observer: "sumof_txndelay [365 0]"
observer>
```

Figure 6.5.2 pow low transaction rate

6.6. Conclusion

In this chapter, we perform a cross-validation of our analytical model and the existing proof-of-stake (PoS) and (Pow) Simulation. The objective is to demonstrate the consistency between the results obtained from our designed analytical model and those generated by the PoS and Pow simulation in our established model. Through rigorous testing, we affirm that our analytical model precisely mirrors the outcomes of the simulation. Specifically, it corroborates that a higher volume of transactions corresponds to an elevated latency delay, while a lower transaction count results in reduced latency delay. This validation reinforces the accuracy and reliability of both our analytical framework and the PoS simulation, crucial for ensuring the fidelity of our blockchain model under varying transaction loads.

CHAPTER 7

Conclusion and Future Work

7.1. Chapter Outline

In this chapter, thesis is concluded with a short summary of all the work done in this research. Conclusion of this research is provided along with a pointer towards the challenges faced and future direction of this research. All possible future work of this research is listed.

7.2. Conclusion

This thesis embarked on a comprehensive exploration of blockchain consensus protocols, conducting a meticulous analysis of their strengths and weaknesses. The critical aspect of latency was addressed through the design of analytical models tailored for each protocol. Sequences and map diagrams were crafted, unraveling the intricate workings of each consensus mechanism. The creation of a blockchain-based simulation model, specifically for testing the Proof-of-Stake (PoS) and Pow consensus protocol, marked a significant milestone. The simulation outcomes were rigorously compared with analytical predictions, revealing a remarkable alignment. This study not only contributes to a nuanced understanding of consensus mechanisms but also establishes the reliability of our analytical frameworks.

7.3. Future Work

Expanding upon this research, future endeavors could encompass a broader spectrum of consensus protocols beyond Proof-of-Stake (PoS). The analytical models, initially tailored for PoS, can be adapted and refined to suit Proof-of-Work (PoW), Delegated Proof-of-Stake (DPoS), and other emerging consensus mechanisms. Furthermore, the integration of smart contracts and decentralized applications (DApps) into the simulation environment can provide a holistic evaluation of consensus protocols in a decentralized ecosystem. Delving into scalability issues and exploring solutions to accommodate an ever-growing number of transactions will be paramount for the sustainability of blockchain networks. Consideration for the environmental impact of consensus mechanisms, especially in the context of energy-intensive PoW, opens avenues for eco-friendly protocols. Collaborative efforts with industry partners can facilitate the validation of these models in real-world blockchain implementations. Lastly, continuous adaptation of models to evolving technological landscapes, such as quantum-resistant consensus, positions this research at the forefront of blockchain innovation.

References

[1] Kuzlu, M., Pipattanasomporn, M., Gurses, L., & Rahman, S. (2019, July). Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In *2019 IEEE international conference on blockchain (Blockchain)* (pp. 536-540). IEEE..

[2] Wan, L., Eyers, D., & Zhang, H. (2019, July). Evaluating the impact of network latency on the safety of blockchain transactions. In *2019 IEEE International Conference on Blockchain (Blockchain)* (pp. 194-201). IEEE.

[3] Wahab, A., & Mehmood, W. (2018). Survey of consensus protocols. *arXiv preprint arXiv:1810.03357*..

[4] Yang, L., Wang, X., Bagaria, V., Wang, G., Alizadeh, M., Tse, D., ... & Viswanath, P. (2019). Practical Low Latency Proof of Work Consensus. *arXiv e-prints*, arXiv-1909.

[5] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017, June). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)* (pp. 557-564). Ieee.

[6] Gracy, M., & Jeyavadhanam, B. R. (2021, November). A Systematic Review of Blockchain-Based System: Transaction Throughput Latency and Challenges. In *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)* (pp. 1-6). IEEE..

[7] Koteska, B., Karafiloski, E., & Mishev, A. (2017, September). Blockchain

implementation quality challenges: a literature. In *SQAMIA 2017: 6th workshop of software quality, analysis, monitoring, improvement, and applications* (Vol. 1938, pp. 8-8).

[8] Fitzi, M., Ga, P., Kiayias, A., & Russell, A. (2018). Parallel chains: Improving throughput and latency of blockchain protocols via parallel composition. *Cryptology ePrint Archive*

[9] Wilhelmi, F., Barrachina-Muñoz, S., & Dini, P. (2022). End-to-end latency analysis and optimal block size of proof-of-work blockchain applications. *IEEE Communications Letters*, *26*(10), 2332-2335..

[10] Hari, A., Kodialam, M., & Lakshman, T. V. (2019, April). Accel: Accelerating the bitcoin blockchain for high-throughput, low-latency applications. In *IEEE infocom 2019-IEEE conference on computer communications* (pp. 2368-2376). IEEE..

[11] Alrubei, S. M., Ball, E. A., Rigelsford, J. M., & Willis, C. A. (2020). Latency and performance analyses of real-world wireless IoT-blockchain application. *IEEE sensors journal*, *20*(13), 7372-7383.

[12] Zhang, S., & Lee, J. H. (2020). Analysis of the main consensus protocols of blockchain. *ICT express*, *6*(2), 93-97.

[13] APEH, A. J., AYO, C. K., & ADEBIYI, A. (2021). A latency-improved blockchain implementation model for nation-wide electronic voting system. *Journal of Theoretical and Applied Information Technology*, *99*(22).

[14] Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N. N., & Zhou, M. (2019). Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism. *IEEE Access*, *7*, 118541-118555.

[15] Sharma, A., & Bhuriya, D. (2019). Literature review of blockchain technology. *IJRAR-Int J Res Anal Rev*, *6*(1).

[16] Sri, P. S. G. A., & Bhaskari, D. L. (2018). A study on blockchain technology. *Int J Eng Technol*, *7*(2.7), 418-421.

[17] Belfer, R., Kashtalian, A., Nicheporuk, A., Markowsky, G., & Sachenko, A. (2020). Proof-Of-Activity Consensus Protocol based on a Network's Active Nodes. CEUR Workshop Proceedings [University Publisher].

[18] Wu, Y., Song, P., & Wang, F. (2020). Hybrid consensus algorithm optimization: A mathematical method based on POS and PBFT and its application in blockchain. *Mathematical Problems in Engineering*, *2020.*

[19] Wang, K., & Kim, H. S. (2022, December). Consensus latency of PoW blockchains. In *2022 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 137-147). IEEE.

[20] D. Yafimava, "Blockchain In the Supply Chain: 10 Real-Life Use Cases and Examples | OpenLedger Insights", Openledger.info, 2019. [Online]. Available: https://openledger.info/insights/blockchain-in-the-supply-chain-use-cases-examples/

[21] Alharby M and van Moorsel A (2020) “BlockSim: An Extensible Simulation Tool for Blockchain Systems.” Front. Blockchain 3:28.doi: 10.3389/fbloc.2020.00028

[22] Catt, M. (2019, March 31). Blockchain Fundamentals: Latency &amp; capacity - featuring the ark ecosystem. Medium. Retrieved July 13, 2022, from https://medium.com/ku-blockchain-institute/blockchain-fundamentals-featuring-the-ark-ecosystem-part-1-af1f9052e579

[23] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., &amp; Qijun, C. (2017, October 1). A review on consensus algorithm of Blockchain: Semantic scholar. . Retrieved July 13, 2022, from https://www.semanticscholar.org/paper/A-review-on-consensus-algorithm-of-blockchain-Mingxiao-Xiaofeng/0bf2cb4ae68275f4fd71a30f191dc95793a0d49e

[24] Castello X, Baronchelli A and Loreto V, Consensus and ordering in language dynamics, 2009 Eur. Phys. J.B 71 55

[25] BlockSim: An Extensible SimulationTool for Blockchain Systems Maher Alharby 1,2* and Aad van Moorsel

[26] B.Kris and C.Decker, Certified Blockchain Business Foundations (CBBF) Official Exam Study Guide v1.1. Blockchain Training Alliance, Inc., 2019

[27] F. Casino, T.K. Dasaklis, C. Patsakis, “A systematic literature review of blockchain-based applications: current status, classification and open issues”, Telematics Inform., 36 (2019), pp. 55-81.

[28] Binance Academy, What is a Blockchain Consensus Algorithm?,https://www.binance.vision/Blockchain/what-is-a-Blockchain-consensus-algorithm

[29] A.Baliga,Understanding Blockchain Consensus Models,2017.Available online:https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf.(Accessed 4 April 2018)

[30] Alharby, M., &amp; van Moorsel, A. (2020, April 28). BlockSim: An Extensible Simulation Tool for blockchain systems. Frontiers. Retrieved July 13, 2022, from https://www.frontiersin.org/articles/10.3389/fbloc.2020.00028/full

[31] Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., &amp; Qijun, C. (2017, October 1). A review on consensus algorithm of Blockchain: Semantic scholar. Retrieved July 13, 2022, from https://www.semanticscholar.org/paper/A-review-on-consensus-algorithm-of-blockchain-Mingxiao-Xiaofeng/0bf2cb4ae68275f4fd71a30f191dc95793a0d49e