

Enhancing the performance of Blockchain-Based IoT Systems



By

Sobia Kanwal

00000318571

Supervisor

Dr. Mian M. Waseem Iqbal

Department of Computer Software Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Masters
in Computer Software Engineering (MS SE)

In

Military College of Signals (MCS) ,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(Aug 2023)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by NS Sobia Kanwal, Registration No. 00000318571, of Military College of Signals has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, 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 and local evaluators of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor Dr. Mian M Waseem Iqbal

Date: 21/8/23

Signature (HOD): _____
Head of Dept of CSE
Mil College of Sigs (NUST)

Date: 22/8/23

Signature (Dean/Principal) _____
Dean, MCS (NUST)
(Asif Masood, Phd)

Date: 13/9/23

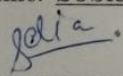
Dedication

This thesis is dedicated to all the deserving children who do not have access to quality education especially young girls.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at Department of Computer Software Engineering at Military College of Signals (MCS) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at Military College of Signals (MCS) or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Sobia Kanwal

Signature: 

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. There is nothing which can payback for His bounties throughout my research period to complete it successfully.

Sobia Kanwal

Contents

1	Introduction and Motivation	1
1.0.1	Blockchain.....	1
1.0.2	Internet of Things (IoT)	2
1.0.3	Blockchain and IoT Systems.....	2
1.0.4	Understanding the Current Consensus Protocols in Blockchain	4
1.0.5	Challenges Faced by IoT Systems in Blockchain.....	5
1.0.6	The Double Spend Problem	7
1.0.7	PLEDGE - A Consensus Protocol Based on Honesty	8
1.0.8	Enhancing IoT Performance with PLEDGE.....	8
1.1	How Does PLEDGE Solve the Double Spend Problem.....	10
2	Literature Review	12
3	Design and Methodology	20
3.1	PLEDGE	20
3.1.1	Random Algorithm.....	20
3.1.2	Verifiable Secret Sharing (VSS).....	21
3.1.3	Delegate Proof-of-Secret Sharing (DPoSS).....	21

CONTENTS

3.2	Improved Network Performance	21
3.2.1	Reduced Communication Overhead	22
3.2.2	Energy Efficiency	22
3.2.3	Scalability	22
3.2.4	Fault Tolerance	23
3.3	How PLEDGE Works	23
3.3.1	Node Attributes and Evaluation	24
3.3.2	Reputation Calculation	24
3.3.3	Consensus Process	24
3.3.4	Best Node Selection	24
3.3.5	Rewards and Penalties	25
3.3.6	Dynamic Reputation Updates	25
3.4	Case Study: Supply Chain Transparency and Traceability	25
3.4.1	Benefits of the Proposed System in Supply Chain Management ..	26
3.4.2	User Roles For Supply Chain Management	27
3.4.3	Network Diagram	28
4	Implementation and Results	30
4.1	Setting Up the Environment	30
4.2	Implementation/Programming	30
4.3	Algorithm	33
4.4	Database Design Of The System	35
4.5	Visual Representation Of the Developed System	36
4.6	Time Evaluation Of Nodes	36

CONTENTS

4.7	Challenges.....	36
4.7.1	Live System Changes.....	36
4.7.2	Debugging.....	37
4.7.3	Minimal Help Online.....	37
4.7.4	Accessing APIs Inside the Code.....	38
4.7.5	Distributed Nature Of The Blockchain.....	38
5	Conclusion and Future Work	39
5.1	Conclusion.....	39
5.2	Future Work.....	39
5.3	Code Availability.....	40
A	Environment	44
A.1	Development Environment Setup.....	44
A.1.1	Set Up Ubuntu.....	44
A.1.2	Setting Up Necessary Tools.....	44
A.2	Ethereum Environment.....	47
A.2.1	Create Directory For the Nodes.....	47
A.2.2	Create Nodes.....	47
A.2.3	Creating Password Files for Accounts.....	48
A.2.4	Create Genesis File.....	48
A.2.5	Initialize All the Nodes With the Genesis File.....	49
A.2.6	Creating Script For Running The Nodes.....	52
A.2.7	Run The Nodes.....	53
A.2.8	Forming The Network.....	53

CONTENTS

A.2.9 Authorizing Other Nodes

A.2.10 Running Delve

A.3 Smart Contract

List of Figures

3.1	Network Diagram For Supply Chain Management	29
4.1	Flow Chart Of PLEDGE	32
4.2	ER Diagram of the Honesty Matrix	35
4.3	Representation Of The Developed System	36
4.4	Representation Of The Developed System	37
A.1	Genesis File A	49
A.2	Gensis File B	50
A.3	GInitialize Geth Node	51
A.4	Delve Error	

List of Tables

2.1	Summary of Literature Review.....	16
2.2	Summary of Literature Review.....	17

Abstract

Although there has recently been a lot of work done for the Blockchain-based IOT systems, these systems still have a few problems. In this document we have taken Ethereum and designed a new honesty based consensus algorithm which aims at optimizing the performance of blockchain based IOT Systems. This protocol selects block signers nodes based on an honesty score which is calculated repeatedly to create a fair an secure system.

CHAPTER 1

Introduction and Motivation

1.0.1 Blockchain

Blockchain is distributed and decentralized ledger technology that enables secure and transparent record-keeping of digital transactions. It consists of a blockchain, each containing a set of transactions. These transactions are verified by a network of computers (nodes) through a consensus mechanism, e.g. proof-of-work or proof-of-stake.

Key features of blockchain include:

- **Decentralization:** The blockchain network operates without a central authority, as it is maintained by a distributed network of nodes.
- **Transparency:** All transactions stored on the blockchain are transparent and can be accessed by anyone on the network. However, the identity of the participants may remain anonymous.
- **Security:** Once a transaction is stored by the blockchain, it is very difficult to alter or tamper with it due to the cryptographic principles used in block creation.
- **Immutability:** The data stored on the blockchain is immutable, meaning it cannot be changed retroactively without the consensus of the network.

Blockchain technology has primarily been associated with cryptocurrencies like Bitcoin, but its potential applications extend beyond financial transactions. It can be utilized for supply chain management, identity verification, smart contracts, voting systems, and more.

1.0.2 Internet of Things (IoT)

The Internet of Things refers to a network of physical items that are equipped with sensors, software, and connection to collect and exchange data. These things might be anything from cellphones and home gadgets to industrial machines and environmental sensors.

Key characteristics of IoT systems include:

- **Connectivity:** IoT devices are internet-connected and may communicate with other devices and systems, forming a network.
- **Data collection and analysis:** Sensors on IoT devices acquire massive volumes of data from their surroundings. This data can be analysed to gain insights and make better decisions.
- **Automation and control:** IoT devices can be remotely controlled and automated based on the data they collect. This allows for efficient monitoring, optimization, and control of various processes.
- **Real-time responsiveness:** To ensure timely actions based on data received, IoT systems frequently demand real-time or near-real-time responsiveness.

1.0.3 Blockchain and IoT Systems

In recent years, the Internet of Things has emerged as a revolutionary technology that enables seamless connectivity and data exchange between various devices. With the increasing adoption of IoT systems in various domains such as smart homes,

healthcare, transportation, and energy management, there is a growing concern about the security and privacy of IoT data. To address these concerns, blockchain technology has emerged as a promising solution that offers secure and decentralized data storage and verification. With recent technological advancements, blockchain has emerged as one of the most promising secure technologies for IoT applications (Subahi & Bouazza, 2020). Blockchain technology, initially introduced as the underlying technology for cryptocurrencies, has shown great potential in enhancing the security and privacy of IoT systems. Blockchain technology provides high security and privacy for different IoT applications and transactions, ensuring data integrity and authenticity [1]. Moreover, blockchain technology offers a high level of management for IoT systems through the use of privileged digital identities and access management [1]. The conventional approaches and reference frameworks for IoT network implementation often fail to meet the stringent security requirements of IoT systems. As a result, researchers and experts have started exploring the potential of integrating blockchain technology into IoT systems to enhance their security and performance [1]. With the integration of blockchain technology, many of the security challenges associated with IoT systems have been addressed. Blockchain and the Internet of Things (IoT) are two transformative technologies that have gained significant attention in recent years. They have the potential to revolutionize various industries and reshape the way we interact with technology. Let's explore a brief introduction to both blockchain and IoT systems and how they can work together. Integrating blockchain with IoT systems can provide several benefits, including enhanced security, data integrity, and trust among participants. Blockchain can serve as a decentralized and tamper-proof ledger for recording IoT device transactions and data exchanges. It can also enable secure and automated smart contracts between IoT devices, ensuring that agreed-upon conditions are met. Additionally, blockchain can address challenges related to data privacy and ownership in IoT systems. By giving individuals more control over their data through blockchain-based identity verification and permission frameworks, users can choose how their data is shared and monetized. Overall, the combination of blockchain and IoT

holds immense potential to create innovative applications and improve various industries, from supply chain management and logistics to healthcare, energy, and smart cities. The decentralized and secure nature of blockchain, when coupled with the data-rich environment of IoT, opens up new possibilities for efficiency, transparency, and trust in the digital era.

1.0.4 Understanding the Current Consensus Protocols in Blockchain

To ensure integrity and dependability of data stored in a blockchain-based IoT system, a consensus mechanism is essential. A consensus mechanism is a protocol that allows all the participants in the blockchain network to agree on the validity of transactions and reach a consensus on the state of the blockchain. Several consensus mechanisms have been proposed and implemented in blockchain networks, each with its own set of characteristics, advantages, and limitations.

Proof of Work, Proof of Stake, and Direct Acyclic Graph are three commonly used consensus mechanisms in blockchain networks. Proof of Work is the most well-known and widely used consensus mechanism in blockchain systems, as it was the original one proposed by Bitcoin for maintaining the security and integrity of its decentralized ledger. Proof of Work necessitates network participants, known as miners, solving challenging mathematical challenges in order to validate transactions and add blocks to the blockchain. Proof of Stake, on the other hand, is an alternative consensus mechanism that does not necessitate miners to solve resource-intensive puzzles. Instead, participants are chosen to validate transactions based on the number of coins they hold and stake in the network. Direct Acyclic Graph is another consensus mechanism that has gained attention in recent years.

It offers a different approach to achieving consensus by structuring the blockchain as a directed acyclic graph rather than a linear chain. The use of a consensus mechanism in blockchain-based IoT systems plays a vital role in ensuring the integrity, security, and performance of the network [2]. Investigating the Need for a New Proof-of-Honesty

Consensus Protocol The existing consensus mechanisms, while effective in certain scenarios, may not be sufficient to address the specific requirements and challenges of blockchain-based IoT systems.

1.0.5 Challenges Faced by IoT Systems in Blockchain

Even if the integration of IoT systems and blockchain, especially Ethereum, has considerable promise for a variety of applications, there are a number of problems that still need to be overcome. In relation to blockchain and Ethereum, the following challenges are typically faced by IoT systems:

Scalability

IoT devices generate vast volumes of data when combined with blockchain, which must be processed and stored by several network nodes. Ethereum is one of the blockchain networks that has scalability concerns. Due to block size limitations and limited transaction processing power, the blockchain's inability to smoothly integrate IoT-generated data may cause congestion and raise transaction fees.

Network Bandwidth and Latency

IoT devices frequently have low network bandwidth and may function in contexts with limited resources. The requirement for the blockchain to transport data across numerous nodes and store it there might put additional strain on network resources, increasing latency and decreasing the efficiency of IoT connection.

Data security and privacy

IoT systems gather sensitive data, such as user information and device telemetry, which may need to be kept secret and confidential. Although data immutability and transparency are intrinsic benefits of blockchain technology, protecting data privacy

and confidentiality is difficult. To protect sensitive IoT data while using the advantages of the blockchain, it is necessary to put in place encryption, access control systems, and off-chain storage options.

Energy Efficiency

IoT devices are frequently fuelled by finite sources of energy, such as batteries or energy harvesting technologies. Integrating blockchain, which relies on resource-intensive consensus algorithms, can significantly impact the energy efficiency of IoT systems. Finding a balance between maintaining blockchain security and minimizing the energy consumption of IoT devices is an ongoing challenge.

Standardisation and Interoperability

There are many different types of IoT platforms, protocols, and devices, which makes interoperability difficult. To enable seamless interoperability between various IoT devices and blockchain networks, integrating blockchain and Ethereum into IoT systems necessitates the establishment of standardised protocols, data formats, and communication interfaces.

Oracles and Data Integrity

Off-chain data for smart contracts is provided by Oracles, which are external data sources used by IoT systems. To avoid data manipulation or tampering, it is essential to maintain the integrity and dependability of these oracles because their compromised data can have a substantial impact on the precision and reliability of smart contract execution.

Cost and Resource Restraints

Running a node's processing needs and storage needs can be resource-intensive aspects of deploying and maintaining blockchain infrastructure. Participating in blockchain networks may provide difficulties for IoT devices with constrained processing power and storage capacity in terms of cost, resource utilisation, and operational viability.

The IoT and blockchain communities must continue to work together on research, development, and other initiatives to meet these problems. Innovative approaches are being investigated to solve these challenges and realise the full potential of integrating IoT devices with blockchain, including Ethereum, such as layer-two scaling strategies, optimised consensus algorithms, privacy-enhancing technologies, and standardisation initiatives.

1.0.6 The Double Spend Problem

Blockchain's "double spend problem" is the potential for a cryptocurrency to be used more than once. When someone has the ability to modify the blockchain network and insert a unique block that enables them to recover spent funds, it happens. If particular requirements are satisfied and the updated blocks are permitted to enter the blockchain, then this may occur. If successful, the initiator of the change can effectively return the cryptocurrency they previously spent and utilise it once more.

Bitcoin and other cryptocurrency blockchains employ a number of security techniques to address the double spend issue:

- **Validation:** A maximum number of network nodes validate transactions. Following the creation of a block, users send validation for the block and it is added to a list of pending transactions. The block is only uploaded to the blockchain after the verifications have been completed.
- **Transactions that have been confirmed are timestamped, making them irreversible.** A Bitcoin transaction that has been validated and confirmed cannot be undone.

Future attempts to conduct more transactions with the same Bitcoin will result in its cancellation.

- **Block Confirmations:** To prevent double spending, merchants receive block confirmations. A minimum of 6 confirmations are needed in Bitcoin.
- **Saving Copies:** Each node retains a copy of each transaction, preventing the loss of the entire network in the event of a network failure.

1.0.7 PLEDGE - A Consensus Protocol Based on Honesty

A sort of consensus algorithm used in distributed systems to reach consensus across numerous nodes is the pledge-based consensus protocol. Each participating node agrees to abide by a set of guidelines or policies in this protocol. The consensus among the nodes is then ascertained using these pledges. A pledge-based consensus protocol's major goal is to ensure the participating nodes' honesty and integrity by compelling them to publicly swear that they will abide by the set of rules. This promotes good behavior and guarantees the distributed system's security and dependability. In many applications, such as blockchain networks, where obtaining consensus among several nodes is essential for maintaining the integrity of the network, the pledge-based consensus protocol can be employed.

1.0.8 Enhancing IoT Performance with PLEDGE

PLEDGE Consensus Protocol: The PLEDGE Consensus Protocol is an IoT-specific distributed consensus mechanism. By tackling the issues raised by resource-constrained devices and network restrictions, it seeks to enhance the performance and scalability of IoT systems.

PLEDGE Consensus Protocol

The PLEDGE Consensus Protocol is an IoT-specific distributed consensus mechanism. By tackling the issues raised by resource-constrained devices and network restrictions, it seeks to enhance the performance and scalability of IoT systems.

Reduced Communication Overhead

The PLEDGE Consensus Protocol helps the Internet of Things run better by lowering communication overhead. In order to accomplish this, the protocol reduces the volume of messages sent and received by IoT devices while the consensus process is taking place. Performance is improved overall as a result of this optimization's reduction of latency and contribution to network bandwidth conservation.

Energy Efficiency

IoT devices frequently use a small amount of battery life. The PLEDGE Consensus Protocol has techniques to improve energy efficiency in light of this. The protocol allows IoT devices run for longer periods of time without needing regular recharging or replacement by lowering the communication overhead and minimising the processing requirements.

Scalability

Internet of Things (IoT) networks may include thousands or even millions of devices. When used in such extensive deployments, the PLEDGE Consensus Protocol is built to scale effectively. IoT devices are arranged into groups using a hierarchical structure, and group leaders are in charge of collecting and transmitting consensus findings. With less computational and communication overhead, the protocol can manage substantial IoT installations while retaining performance.

Fault Tolerance

Device malfunctions and network interruptions are common in IoT networks. By enabling continuing operation despite faults, this fault tolerance feature improves the overall performance of the IoT system.

1.1 How Does PLEDGE Solve the Double Spend Problem

- When several competing transactions are transmitted at the same time in a decentralized network, it can be difficult to decide which transaction should come first. When someone successfully uses the same digital currency twice, it creates a problem known as double spend.
- Like other consensus methods, the PLEDGE Honesty-based Consensus Protocol seeks to organize transactions and avoid double-spending. It accomplishes this by designing a system that generates a network's canonical order of occurrences.
- Each participant presents their account of the events as block candidates, and the protocol uses a fair selection technique to select the winners. The shared truth ultimately emerges through the selection of a winning block. Only one version of the transaction is accepted as a result of this process, which also destroys any ties between conflicting transactions.
- The selection process in the PLEDGE Honesty-based Consensus Protocol extends proof-of-authority (PoA) mechanism. By employing processing power to attempt to crack challenging mathematical puzzles, miners take part in the fair selection. The next block will be generated and added to the blockchain by the first miner to crack the mystery. The technique encourages miners to work together and build on one another's blocks, making attempts at double-spending difficult from an economic standpoint.

CHAPTER 1: INTRODUCTION AND MOTIVATION

- The protocol makes sure that blocks are accepted in the blockchain in a decentralized and secure manner by employing PoW as the fair selection mechanism. The transactions become probabilistically immutable when more blocks are added because they are buried by a mountain of proof-of-work. This means that it is very challenging to change or undo a transaction once it has been incorporated into a block and the blockchain.
- Like other consensus protocols, the PLEDGE Honesty-based Consensus Protocol depends on network-wide involvement to maintain the rules. The other nodes in the network will reject double spends, hence miners, as the writers of block candidates, must not include them in their blocks.

The double spend issue is resolved by implementing the PLEDGE Honesty-based Consensus Protocol by establishing a shared truth using a fair selection process and making sure that blocks are uploaded to the blockchain in a secure and decentralised way.

CHAPTER 2

Literature Review

[3] Challenges in handling IoT and blockchain integration. An examination of the possible benefits of blockchain for IoT. Blockchain IoT apps and platforms for IoT solution creation. Examine current blockchain-IoT systems and applications. Topologies for that integration could be considered. Blockchain node evaluation in IoT devices[4].

Because blockchains are primarily systems for worldwide shared trust, they are extremely powerful technologies beyond simple security applications. This article illustrates certain IoT scenarios in which BCMs play a key role while emphasizing that BCMs are merely a component of the IoT Security (IoTSec) solution. However, it may not be possible due to the normal constraints of IoT nodes. [5]

Internet of Things is a well-known computing technology concept. It is increasingly being used to help people live better lives through a number of applications, of which smart healthcare, smart cities, smart grids, and smart finance are just a few examples. Scalability, interoperability, security, privacy, and trustworthiness are all challenges for IoT systems. Blockchain solutions have recently been created to help overcome these obstacles. The significance of blockchain technology is highlighted in terms of features and benefits for IoT application components. [6] have proposed a blockchain taxonomy for IoT applications based on the most relevant elements. Furthermore, they have

investigated the most extensively used blockchain platforms for IoT, examined how blockchain technology can be utilized to increase the range of IoT applications. In addition, they have covered new advancements and solutions for IoT contexts and discussed the challenges and potential research areas of using blockchain for IoT.

The main challenges of implementing IoT in agriculture have been categorized into 6 different categories. The first challenge is the selection of the appropriate sensor from different varieties (e.g., temperature, pressure, proximity, water quality, humidity, and so forth...) that provide different kinds of metrics. Another challenge is the implementation of different predictive techniques in the IoT application qualified as the data analytics challenge. The maintenance challenge involves the maintenance required by the sensors used. The type of wireless technology used is another challenge i.e. the mobility challenge. Installation of the IoT infrastructure using modern technologies such as fog computing, and network virtualization is also a challenge. The main challenge however is security and privacy. Implementation of IoT in agriculture may give rise to more security and privacy risks. Similarly, various papers (i.e. [7], [8], [9], [10], [11]) focus on the attributes of IoT-based agricultural applications but none of them cater to privacy and security-related research challenges. Cha et al. [12] give an account of different privacy-enhancing technologies in IoT and have categorized them into 7 categories. Security requirements are mainly classified into 5 major categories which should be the prerequisites for security protocols in IoT [13]. The categories are authentication, non-repudiation, confidentiality, access control, and integrity. Ferrag et al. [14] have analyzed surveys conducted in the field of IoT security by four major checks, namely Threat Model - whether it took into account the challenges to IoT networks, security, and privacy - whether the measures to protect the IoT network, Blockchain - whether blockchain-based IoT Applications were considered and Target IOT Application - whether specific or general. As the first survey to address all of these checkpoints for green IOT-based farm applications. Significant challenges, however, persisted in machine learning methodologies, datasets for intrusion detection, scalability analysis of blockchain-based solutions, choosing the best consensus

algorithm, and building practical and compatible cryptographic protocols. A consensus protocol capable of validating transactions without the help of a reliable third party was made public by Bitcoin. But why is this technique so incredible? Its peer-to-peer nature, which is a completely decentralized protocol that enables transactions to be securely validated in an open network where users can join and leave at any time, is one of the reasons. Byzantine Fault Tolerant (BFT) consensus mechanisms did in fact exist prior to Bitcoin, but they were restricted to closed networks with a predetermined number of players.[15] A new decentralized and distributed technology is called a blockchain. The consensus algorithms of blockchains serve as important pillars for this technology, which also benefits from decentralization, transparency, and security. A decentralized decision-making process is facilitated by consensus protocols and algorithms. All participants are involved in an inclusive consensus mechanism that bases decisions on conflicts in blockchain networks. These consensus decisions help to achieve finality and improve the quality of blockchain results. To improve or enhance the current consensus protocols, a thorough study is now being conducted. As the current versions of the protocols are unsuitable for resource-constrained environments due to their complexity, difficult configurations, mining techniques, high resource consumption, and explicit security loophole, the optimized or enhanced consensus protocols aim to be suitable for Internet-of-Things (IoT). A survey of consensus protocols with the goal of identifying and debating the presence of various consensus protocols available in the literature was done by[15][16][17][18]. The focus was a specific emphasis on the protocols' origins, particularly Proof-of-X, Byzantine fault tolerance, Paxos, and RAFT. They have discussed the DAG orientation of some current algorithms. In comparison to other surveys in the field, the current survey assists researchers and application developers in gaining insight into the current status of the consensus protocols' suitability to deliver the desired functionalities in IoTs by providing a more thorough summary of the most pertinent protocols and application issues. Each protocol's recognized disadvantages provide future promise for industry and academia. Our claims regarding our contribution are significant since, to the best

CHAPTER 2: LITERATURE REVIEW

of our knowledge, there is no such comprehensive and concise survey of consensus protocols, including DAG-based protocols, in the literature.

CHAPTER 2: LITERATURE REVIEW

Paper	Year	Major Contribution	Comparison With Current Work	Domain
[4]	2018	Surveys blockchain and its integration with IoT, it's challenges and opportunities	Study Challenges	Blockchain in IoT
[5]	2018	Blockchain Security Mechanisms for IoT	Study Security Challenges	Blockchain in IoT
[6]	2022	Taxonomy, Platforms, Recent Advances, Challenges, and Future Research Directions in Blockchain for IoT Applications	Studies Blockchain Implementation For IoT	Blockchain in IoT
[7]	2018	An overview of the Internet of Things (IoT) and data analytics in agriculture: the advantages and disadvantages.	Studies Challenges	Blockchain in IoT
[8]	2017	IoT in agriculture: Creating a large-scale pilot programme across Europe	Studies Challenges	Blockchain in IoT
[9]	2017	Smart agriculture with the Internet of Things: Technologies, Practises, and Future Directions	IoT Implementation in Agriculture	IoT
[10]	2019	Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture	IoT Implementation in Agriculture	IoT
[11]	2019	A life cycle framework for green IoT-based agriculture, including issues of finance, operation, and management.	IoT Implementation in Agriculture	IoT

Table 2.1: Summary of Literature Review.

CHAPTER 2: LITERATURE REVIEW

Paper	Year	Major Contribution	Comparison With Current Work	Domain
[11]	2018	Perspectives and problems of privacy-enhancing technologies in the Internet of Things	IoT Privacy Enhancement	IoT
[13]	2017	A comprehensive study of authentication techniques for the internet of things.	IoT Authentication Protocols	IoT
[14]	2020	Review, blockchain solutions, and difficulties for green IoT-based agriculture security and privacy	IoT Security & Privacy	IoT in Blockchain
[15]	2022	Blockchain consensus protocols, from Bitcoin to Ethereum 2.0, in a nutshell	Studies Blockchain Consensus Protocols	Blockchain
[16]	2022	A comparison of hyper ledger fabric and ethereum in the medical sector: A systematic review and studies Comparison of Ethereum and Hyperledger Fabric	Blockchain	
[17]	2022	Two GHOST/Ethereum Proof-of-Stake Attacks	Studies Security of Proof Of Stake Consensus Algorithm	Blockchain
[18]	2022	A overview and classification of blockchain consensus protocols	Studies Protocols for Blockchain Consensus	Blockchain

Table 2.2: Summary of Literature Review.

Moreover many consensus protocols have been proposed that are written specifically for IOT based blockchain networks. Some of them are briefly introduces below:

- **MBFT:** For consortium blockchain, MBFT a new consensus algorithm [19] With this technique, consortium blockchains will have a Byzantine Fault Tolerant (BFT) consensus mechanism. It seeks to guarantee the network's dependability and security.
- **Proteus:** Proteus is a blockchain-compatible BFT consensus protocol. [19] Scalability is the main objective of Proteus, another BFT consensus protocol. In order to enable larger-scale blockchain networks, it tries to address the shortcomings of conventional BFT protocols.
- **PoBT:** A compact consensus mechanism for IoT business blockchains [19] A lightweight consensus mechanism called PoBT was created exclusively for scalable IoT business blockchains. It seeks to alleviate IoT device resource limitations while assuring effective consensus.
- **PPCoin:** Peer-to-peer cryptocurrency PPCoin uses proof-of-stake. Cryptocurrency PPCoin employs the proof-of-stake consensus mechanism. Despite not being specifically created for IoT-based blockchains, it is nonetheless worth taking into account because to its energy-efficient design.
- **Blockchain consensus approaches are compared[20]** This paper offers a thorough examination of the different consensus methods employed in blockchain networks. It can be a useful tool for figuring out the advantages and disadvantages of certain algorithms.
- **A comprehensive analysis of the literature on blockchain consensus protocols[20]** The various consensus protocols utilised in blockchain networks are summarised in this review. It talks about their uses, benefits, and drawbacks, which is useful for comprehending the range of consensus algorithms for IoT-based blockchains.

- **Blockchain industrial applications for IoT data [21]** The handling of IoT data via blockchain is explored in this study. Although it is not explicitly focused on consensus techniques, it offers insights into the real-world applications of blockchain in the IoT space.
- **A comparison of consensus algorithms in conjunction with blockchain and the Internet of Things [21]** In the context of collaborating IoT and blockchain, this study compares various consensus algorithms. It examines key factors for IoT-based blockchain networks, including performance, scalability, and dependability.

CHAPTER 3

Design and Methodology

Blockchain technology relies on consensus procedures to ensure that all network users agree on the status of a distributed ledger. They set up regulations for confirming transactions and prohibiting harmful activity on the network. To validate transactions, participants in the Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus procedures, must solve challenging problems or hold a specific amount of tokens. Although these protocols are useful, they have several drawbacks, especially when used with Internet of Things (IoT) systems.

3.1 PLEDGE

PLEDGE is specifically created for blockchain-based Internet of Things (IoT) systems with the goal of ensuring participant honesty and integrity. Let's examine the PLEDGE protocol's salient characteristics and workings:

3.1.1 Random Algorithm

PLEDGE employs a random algorithm to select a set of nodes, known as packers, from the network. This selection process is based on the parameters of all nodes in the network, ensuring fairness in the election . The random algorithm helps prevent

manipulation and ensures that every participant has an equal chance to contribute to the consensus process.

3.1.2 Verifiable Secret Sharing (VSS)

PLEDGE includes verifiable secret sharing to safeguard sensitive data during transmission. The information is divided and encrypted using this method in a way that ensures confidentiality and restricts access. Participants can check the validity and integrity of the shared secrets thanks to the verifiable component. VSS boosts the protocol's general security and its privacy-preserving features.

3.1.3 Delegate Proof-of-Secret Sharing (DPoSS)

Delegate Evidence of PLEDGE employs Secret Sharing as a delegate-based validation technique. A group of nodes referred to as tellers verifies transactions within the network. In contrast to PoW, where all nodes participate in the validation process, DPoSS promotes scalability by delegating the validation task to a specified group of specialised nodes. This method offers a more efficient consensus procedure when there are resource constraints in IoT systems. PLEDGE develops an honesty-based consensus protocol appropriate for blockchain-based IoT systems by fusing the aforementioned components. It uses a Proof-of-Honesty method to ensure that participants behave honestly, verifiable secret sharing for safe data transmission, delegation of validation chores to a small number of nodes for scalability, and exclusion of sensors from voting to optimise the protocol.

3.2 Improved Network Performance

The PLEDGE Consensus Protocol improves IoT performance in the following ways:

3.2.1 Reduced Communication Overhead

The PLEDGE Consensus Protocol minimizes the number of messages exchanged between IoT devices during the consensus process. This optimization helps conserve network bandwidth and reduces latency, resulting in improved overall performance. By reducing communication overhead, the protocol ensures efficient utilization of network resources. To illustrate how the PLEDGE Consensus Protocol reduces communication overhead, consider a scenario where multiple IoT devices need to agree on a particular decision. Without a consensus protocol, each device would have to communicate with all other devices to reach an agreement. This would lead to a significant amount of data exchange, causing network congestion and increased latency. With the PLEDGE Consensus Protocol, devices only need to communicate with a subset of other devices, reducing the overall communication overhead and improving performance.

3.2.2 Energy Efficiency

IoT devices frequently have low battery life. The PLEDGE Consensus Protocol has techniques to improve energy efficiency in light of this. The protocol allows IoT devices run for longer periods of time without needing regular recharging or replacement by lowering the communication overhead and minimising the processing requirements. The PLEDGE Consensus Protocol reduces the computing load on IoT devices to achieve energy efficiency. To minimise the amount of processing required for the consensus process, it uses effective algorithms and optimisation techniques. The protocol lowers the energy consumption of IoT devices, improving energy efficiency and extending device lifetimes by reducing the processing needs.

3.2.3 Scalability

Millions or even thousands of devices may be part of an IoT network. When used in such extensive deployments, the PLEDGE Consensus Protocol is built to scale effectively. IoT devices are arranged into groups using a hierarchical structure, and

group leaders are in charge of collecting and transmitting consensus findings. With less computational and communication overhead, the protocol can manage substantial IoT installations while retaining performance. The PLEDGE Consensus Protocol's hierarchical structure enables effective device coordination and consensus in massive IoT networks. Only group leaders must communicate with one another, greatly decreasing the communication overhead, as opposed to every device taking part in the consensus process. The protocol can accommodate a growing number of IoT devices thanks to its scalability feature without compromising.

3.2.4 Fault Tolerance

Device malfunctions and network interruptions are common in IoT networks. For the consensus process to be dependable and available, the PLEDGE Consensus Protocol includes fault tolerance features. In order to withstand failures at both the device and network levels, it uses redundancy and backup mechanisms. This fault tolerance capability improves the IoT system's overall performance by assuring that it can continue to function even in the face of failures. To guarantee fault tolerance, the PLEDGE Consensus Protocol employs redundancy and backup techniques. Backup devices and other communication channels are accessible to continue the consensus process in the event that a device or network link fails. This redundancy lessens the effects of failures, ensuring that the system continues to function properly and that agreement may still be reached even in difficult situations.

3.3 How PLEDGE Works

A reputation-based consensus mechanism might be implemented in the IoT-based Ethereum network to identify the best nodes for decision-making by comparing the qualities of various nodes. Based on the characteristics, actions, and network contributions of participating nodes, this kind of consensus method evaluates their reputation or performance. An overview of how such a system may operate is given

below:

3.3.1 Node Attributes and Evaluation

Every node in the network is given a certain set of traits or properties that are crucial to its dependability and functionality. Node uptime, processing power, energy efficiency, prior behaviour, responsiveness, and overall network contribution are a few examples of these traits.

3.3.2 Reputation Calculation

Each node's reputation is established based on the attributes offered. The actual formula for calculating reputation may vary depending on the methods employed. Utilising a weighted average of the features, where certain traits are more crucial than others in determining reputation, is one possibility.

3.3.3 Consensus Process

Nodes broadcast their suggested transactions or blocks to the network during the consensus process. Then, other nodes assess these recommendations and take into account the standing of the original nodes. As a result, proposals from nodes with better reputations are given more weight or influence during the consensus process.

3.3.4 Best Node Selection

The consensus method chooses the best nodes for decision-making based on reputation ratings and the appraisal of proposals. These nodes tend to have the best ratings or are the most dependable in the network. When deciding the blockchain's agreed-upon state, their suggestions are given precedence.

3.3.5 Rewards and Penalties

The reputation-based consensus algorithm might have incentives for good behaviour, such as rewards, and disincentives for bad behaviour, such as penalties. Nodes that consistently display favourable characteristics and make excellent contributions to the network may be granted extra rights, such as greater influence over the consensus procedure or greater rewards for their participation.

3.3.6 Dynamic Reputation Updates

Based on a node's current behaviour and contributions to the network, its reputation scores may be dynamically updated. While nodes that perform well can earn higher reputations, nodes with dropping reputations may have less influence over the consensus process.

The IoT-based Ethereum network can give higher priority to nodes with better reputations and characteristics that advance the network's objectives by implementing a reputation-based consensus method. This makes it possible to guarantee that decisions made during the consensus process are affected by nodes that are seen as being more credible, trustworthy, and advantageous to the network as a whole.

3.4 Case Study: Supply Chain Transparency and Traceability

Luxury goods frequently experience challenges with fraud and counterfeiting, such as high-end clothing or luxury watches. Manufacturers and customers may guarantee the authenticity and traceability of these commodities across the supply chain by utilising a blockchain-based IoT system with an honesty-based consensus protocol pledge.

3.4.1 Benefits of the Proposed System in Supply Chain Management

This use case would operate as follows:

- Each opulent item is given a distinctive identifier, such as an RFID chip or a QR code, for secure product identification. These identifiers are read and tracked by IoT devices like sensors and scanners. The Internet of Things (IoT) devices gather pertinent information, including timestamps, locations, and conditions, each time an item is exchanged or passes through a certain event in the supply chain.
- Ledger Built on a Blockchain: The data is then safely kept on a blockchain. An unchangeable ledger provided by the blockchain ensures participant transparency and confidence. On the blockchain, every transfer or occurrence is documented as a transaction, creating a transparent audit trail for the whole lifecycle of the luxury items.
- Honesty-Based Consensus Protocol: An honesty-based consensus protocol can be used to validate and verify the data gathered by IoT devices. According to this protocol, before a transaction could be added to the blockchain, many nodes in the blockchain network would have to agree on its legitimacy and accuracy. Since honesty is a fundamental tenet of the protocol, only authenticated and true data is allowed.
- Product Authentication and Traceability: With the blockchain-based IoT system in place, supply chain participants like producers, distributors, and retailers can quickly trace and confirm the authenticity of luxury goods. They may use the blockchain to check each item's provenance, ownership history, and supply chain events. Transparency promotes consumer trust and aids in the fight against counterfeiting.
- Consumer Verification: Before making a purchase, consumers can use this method to confirm the legitimacy of premium items. They can instantaneously access the product's whole history from the blockchain by scanning the product's unique

identification using mobile applications or websites. Because of this, customers may make informed decisions about their purchases and stay away from fake goods.

The luxury goods sector may make great strides towards ensuring the authenticity, transparency, and traceability of their products throughout the supply chain by integrating blockchain technology with IoT devices and a promise of honesty-based consensus mechanism.

3.4.2 User Roles For Supply Chain Management

For blockchain network to be used to manage the supply chain, and there are a number of user roles that each perform a unique role and have a unique set of duties. These jobs consist of:

- **Supplier:** Suppliers offer the components or raw materials required for the manufacturing process. They can add details to the blockchain network about the materials' provenance, calibre, and delivery.
- **Manufacturer:** Producing the commodities or products that will be a part of the supply chain is the manufacturer's responsibility. They can add data to the blockchain network concerning the manufacturing process, quality assurance, and product specs.
- **Distributor/ Quality Control:** Distributors are in charge of shipping products from the producer to retailers or final consumers. They can enter tracking and delivery information as well as information about the transportation process into the blockchain network. Throughout the supply chain, quality control staff are in charge of assuring the products' conformity and quality. They can upload records of compliance, certificates, and quality inspections to the blockchain network.
- **Logistics Provider:** Within the supply chain, logistics providers manage the transportation and logistical operations. They can add data to the blockchain network regarding the movement of items, storage conditions, and delivery schedules.

- **Retailer:** Businesses that sell goods to final consumers are known as retailers. They can add data to the blockchain network concerning sales, inventories, and consumer feedback.
- **Customers:** Customers are the final consumers or users of the goods. They might have access to details like product validity, provenance, and other pertinent information on the blockchain network.
- **Auditor:** Independent organisations called auditors check and validate the data stored on the blockchain network. To assure the data's correctness and integrity, they could conduct audits.

In order to guarantee transparency, traceability, and effectiveness in the supply chain management process, these user roles collaborate and communicate with one another over the blockchain network. Each position contributes to the overall efficiency of the blockchain-based supply chain system while carrying out specialised duties.

3.4.3 Network Diagram

All the nodes of this network are supposed to maintain an Honesty matrix which is calculated at every level.

All the nodes are connected to a blockchain network with PLEDGE. Here's how it'll look in the deployment.

Supplier supplies the raw materials to the Manufacturer, who manufactures the products, Distributor takes these good performs quality control and ships it for selling through the Logistics. Goods can be shipped directly to the Consumer in a B2C online model and it can be sold through Retailers too. Figure below shows the network diagram by roles:

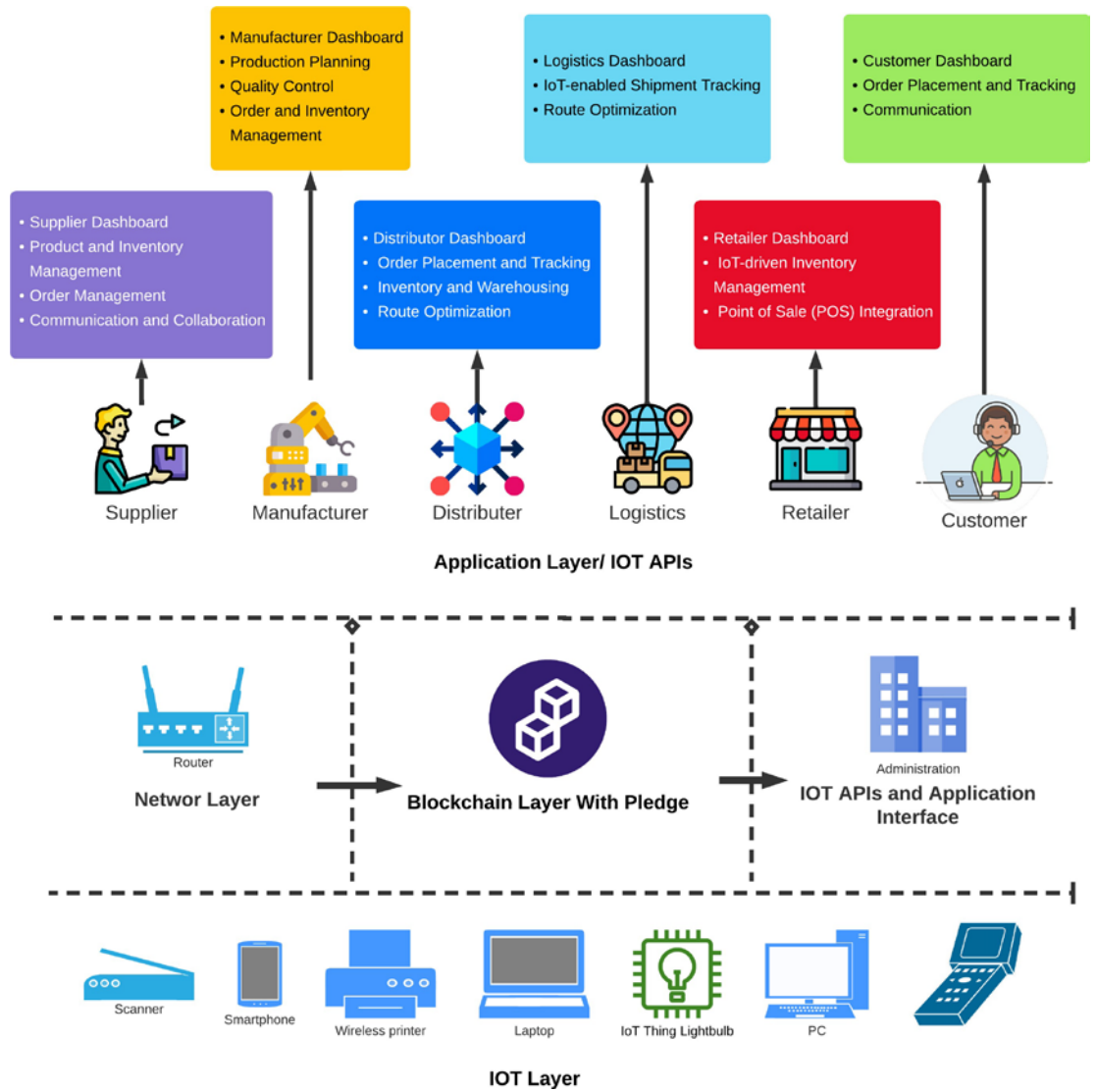


Figure 3.1: Network Diagram For Supply Chain Management

CHAPTER 4

Implementation and Results

4.1 Setting Up the Environment

This system was implemented in Golang using the live code available in the go-ethereum's github repo. The Operating System used was Ubuntu. For the prototype three miner nodes were created using the shell script. And one full node was created. They were connected to each other by adding the enode ID's of all the peer nodes as peers to the first node. Hence a system of total 4 nodes was created. For the system to run, genesis file was created using the puppeth. Puppeth was later removed for some reason. But genesis file can be created manually using the basic configuration settings. Details of Environment setup is given in Appendix A.

4.2 Implementation/Programming

After the environment has been setup, it was time for introducing the new consensus algorithm to the code. We took proof of Authority i.e. Clique as the basis for creating the new algorithm. For that we first analyze the code.code so that we can incorporate the desired changes. The code was operational so we did not want to destroy it by making changes in the wrong places. After carefully analyzing the code, we figured out the places where clique was being used. We then started adding pledge alongside clique

by duplicating or replacing the code. In most of the cases it was easy to make the changes as it could be done alongside clique without breaking any of the functionality of clique but in some cases it was really tricky to handle the code as it was conditional and we had to carefully incorporate our part. After that had been done and all the nodes were working with the newly created consensus algorithm, pledge was also available in the puppet interface, we then moved to the next step which was implementing out PLEDGE algorithm in the code. For that we figured out that the main function had to be written in the pledge.go file in the consensus/pledge module of the code. We wrote the main function in this file and many other files had to be changed consequently for these functions to run. Here is a rough flow chart of the process because the detailed flow chart has already been given in the original paper.

$$\frac{1}{n} \sum_{i=1}^n x_i$$

Where x = Some Of Honesty Matrix of all the Nodes n = Number Of Authorized Signers

Here's a diagram of how the algorithm works.

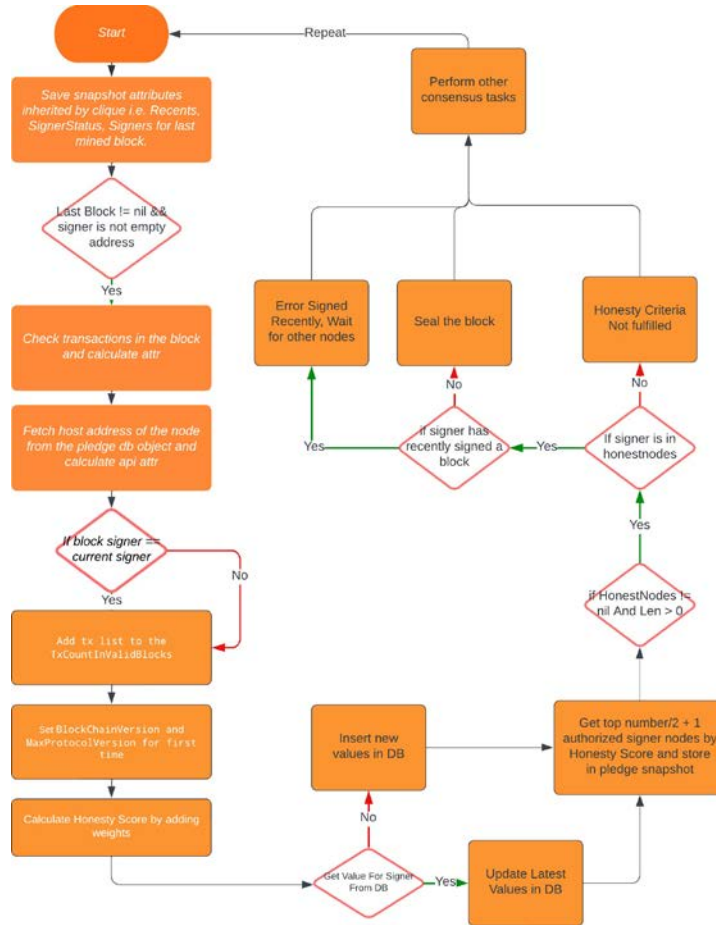


Figure 4.1: Flow Chart Of PLEDGE

4.3 Algorithm

PLEDGE works on the principle of Proof of Authority or basically it can be called an extension of the PoA. Once a signer proposes a block, it is first check for authorization just like Proof of Authority. If the block is authorized by the network, it can then move forward in the process. Next it is check in the recent nodes. If there are more than one signers in the network and it has signed recent, it is skipped and the next signer is processed. The node is removed from recent nodes list. Next it is check for valid network ID and protocol versions. If all are valid, honesty score of the signer is compared with the average honesty score of the network which is computed by dividing the sum of honesty scores of all the nodes by the total number of signers.

CHAPTER 4: IMPLEMENTATION AND RESULTS

Attributes	Weight Criteria	Calculation
TxCountInValidBlocks	1 mk for every TX	Calculated for each signer whenever a new block is proposed
TxSent	1 mk for every TX	Calculated for each signer whenever a new block is proposed
TxReceived	1 mk for every TX	Calculated for each signer whenever a new block is proposed
TxErrors	-10 mk for every error	Calculated for each signer whenever a new block is proposed
ValidBlocksFinalized	1 mk for each block	Calculated for each signer whenever a new block is proposed
PendingTX	-1 mk for every pending TX	Calculated for each signer whenever a new block is proposed
IsListening	-10 mks for not listening 1 for listening	Fetches from the network whenever a new block is proposed
PeerCount	2 mks for each connection	Fetches from the network whenever a new block is proposed
Balance	Divided by 100	Calculated for each signer whenever a new block is proposed
AmountSpent	Divided by 100	Calculated for each signer whenever a new block is proposed
AmountReceived	Divided by 100	Calculated for each signer whenever a new block is proposed
HonestyScore	Sum of all the attribute Values	Calculated for each signer whenever a new block is proposed
ChainId	-10 for incorrect ID 1 for correct ID	Saved once for each node and compared with the latest value
BlockChainVersion	-1 for old ver 1 for latest ver	Saved once for each node and compared with the latest value
MaxProtocolVersion	-1 for old ver 1 for latest ver	Saved once for each node and compared with the latest value

4.4 Database Design Of The System

We used pledge snapshot struct to save the data of the honesty matrix along with the Other Attributes i.e. Signers and Recents etc. And the Current average running in the network. We also have to save it to a database as the memory database saves data of each node seperately. So we need to integrate an SQLite database to store and retrieve the information of the honesty matrix.

<i>Honesty Matrix</i>	
<i>Signer</i>	<i>Address</i>
<i>BlockHash</i>	<i>Hash</i>
<i>BlockIndex</i>	<i>Interger</i>
<i>TxCountInValidBlocks</i>	<i>Interger</i>
<i>TxSent</i>	<i>Interger</i>
<i>TxReceived</i>	<i>Interger</i>
<i>TxErros</i>	<i>Interger</i>
<i>ValidBlocksFinalized</i>	<i>Interger</i>
<i>PendingTx</i>	<i>Interger</i>
<i>IsListening</i>	<i>bool</i>
<i>PeerCount</i>	<i>Interger</i>
<i>Balance</i>	<i>Interger</i>
<i>AmountSpent</i>	<i>Interger</i>
<i>AmountReceived</i>	<i>Interger</i>
<i>HonestyScore</i>	<i>Interger</i>
<i>ChainId</i>	<i>Interger</i>
<i>BlockChainVersion</i>	<i>Interger</i>
<i>MaxProtocolVersion</i>	<i>Interger</i>

Figure 4.2: ER Diagram of the Honesty Matrix

4.5 Visual Representation Of the Developed System

Here is a visual representation of the proposed system. Each miner node will maintain an honesty matrix.

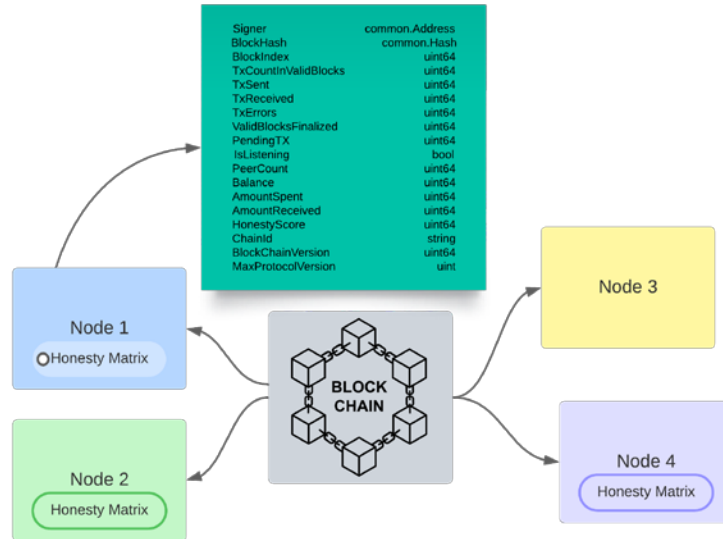


Figure 4.3: Representation Of The Developed System

4.6 Time Evaluation Of Nodes

4.7 Challenges

We had to face many challenges in implementing this system. Some of them are given as follows:

4.7.1 Live System Changes

As the software was operational, there were multiple changes being pushed to the code all the time. Creating a new module along with the continuous changes was a difficult

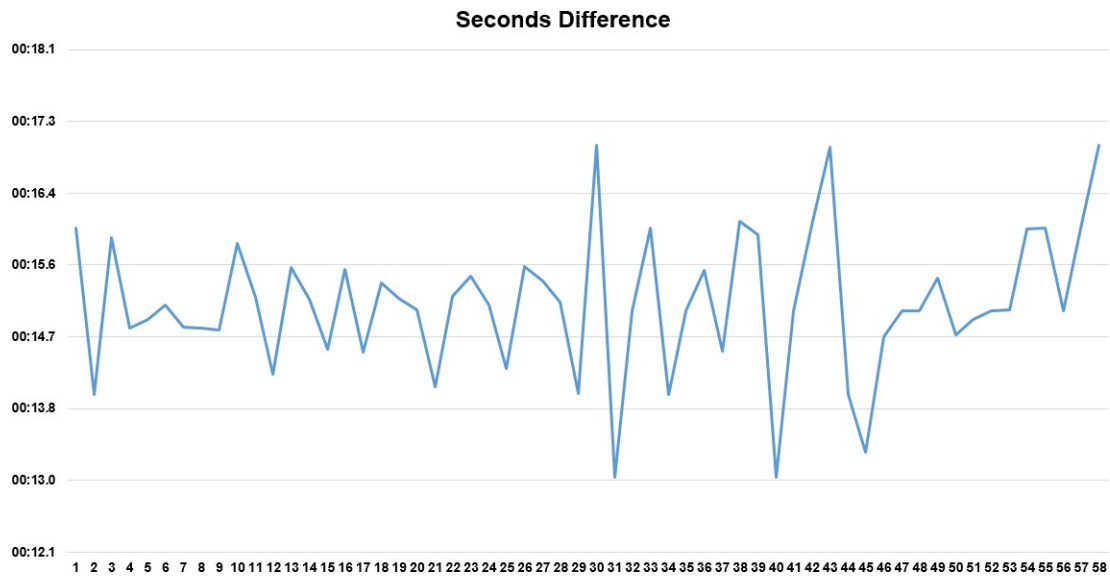


Figure 4.4: Representation Of The Developed System

task as code was being removed and added continuously. The code needed updation all the time and we had to think of new ways to match the updated mosules.

4.7.2 Debugging

Debugging code run by the shell scripts was an issue as it was sometimes or most of the time difficult to catch the error. We had to install logging command to views data which was sometimes not accessible. Also figuring out how to configure the debugger was a challenge.

4.7.3 Minimal Help Online

As this a field currently in research phase, it was difficult to implement without any kind of help available. There was a lot of help for writing smart contracts but making changes in the live code was a little challenging.

4.7.4 Accessing APIs Inside the Code

Accessing javascript APIs inside the code was challenging to figure out so it was a little difficult in calculating some attributes.

4.7.5 Distributed Nature Of The Blockchain

The databases are maintained separately for each node, so we have to install a separate database for maintaining our honesty matrix.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

Overall, the PLEDGE Consensus Protocol improves the performance of the Internet of Things (IoT) by lowering communication overhead, increasing energy efficiency, enabling scalability, and offering fault tolerance. It decreases latency, conserves network bandwidth, and limits the quantity of messages that are sent back and forth between devices. The protocol makes the most efficient use of energy, extending the battery life of IoT devices. Using a hierarchical structure, it scales effectively in big deployments. In order to guarantee availability and reliability, it also includes fault tolerance techniques. However, while choosing a consensus protocol for a specific IoT deployment, it's crucial to take other aspects into account, such as network topology, application needs, and system limits.

5.2 Future Work

PLEDGE can be further optimized by working on the attributes and experimenting with the weights to different attributes.

5.3 Code Availability

The code is currently not available on github. Though the up to date changes are available in the CD.

Bibliography

- [1] Fahad F Alruwaili. “Intrusion detection and prevention in Industrial IoT: A technological survey”. In: *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE. 2021, pp. 1–5.
- [2] Ahmad F Subahi and Kheir Eddine Bouazza. “An intelligent IoT-based system design for controlling and monitoring greenhouse temperature”. In: *IEEE Access* 8 (2020), pp. 125488–125500.
- [3] Imran Makhdoom et al. “PLEDGE: A proof-of-honesty based consensus protocol for blockchain-based IoT systems”. In: *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2020, pp. 1–3.
- [4] Ana Reyna et al. “On blockchain and its integration with IoT. Challenges and opportunities”. In: *Future generation computer systems* 88 (2018), pp. 173–190.
- [5] Daniel Minoli and Benedict Occhiogrosso. “Blockchain mechanisms for IoT security”. In: *Internet of Things* 1 (2018), pp. 1–13.
- [6] Abdelzahir Abdelmaboud et al. “Blockchain for IoT Applications: Taxonomy, Platforms, Recent Advances, Challenges and Future Research Directions”. In: *Electronics* 11.4 (2022), p. 630.
- [7] Olakunle Elijah et al. “An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges”. In: *IEEE Internet of things Journal* 5.5 (2018), pp. 3758–3773.

BIBLIOGRAPHY

- [8] Christopher Brewster et al. “IoT in agriculture: Designing a Europe-wide large-scale pilot”. In: *IEEE communications magazine* 55.9 (2017), pp. 26–33.
- [9] Partha Pratim Ray. “Internet of things for smart agriculture: Technologies, practices and future direction”. In: *Journal of Ambient Intelligence and Smart Environments* 9.4 (2017), pp. 395–420.
- [10] Abhishek Khanna and Sanmeet Kaur. “Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture”. In: *Computers and electronics in agriculture* 157 (2019), pp. 218–231.
- [11] Junhu Ruan et al. “A life cycle framework of green IoT-based agriculture and its finance, operation, and management issues”. In: *IEEE communications magazine* 57.3 (2019), pp. 90–96.
- [12] Shi-Cho Cha et al. “Privacy enhancing technologies in the Internet of Things: Perspectives and challenges”. In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 2159–2187.
- [13] Mohamed Amine Ferrag et al. “Authentication protocols for internet of things: a comprehensive survey”. In: *Security and Communication Networks* 2017 (2017).
- [14] Mohamed Amine Ferrag et al. “Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges”. In: *IEEE access* 8 (2020), pp. 32031–32053.
- [15] Sara Tucci-Piergiovanni. “Keynote: Blockchain consensus protocols, from Bitcoin to Ethereum 2.0”. In: *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE. 2022, pp. 1–1.
- [16] KB Jyothilakshmi, Vandana Robins, and AS Mahesh. “A comparative analysis between hyperledger fabric and ethereum in medical sector: A systematic review”. In: *Sustainable Communication Networks and Application* (2022), pp. 67–86.
- [17] Joachim Neu, Ertem Nusret Tas, and David Tse. “Two Attacks On Proof-of-Stake GHOST/Ethereum”. In: *arXiv preprint arXiv:2203.01315* (2022).

BIBLIOGRAPHY

- [18] Arshdeep Singh et al. “A survey and taxonomy of consensus protocols for blockchains”. In: *Journal of Systems Architecture* (2022), p. 102503.
- [19] Yujuan Wen et al. “Blockchain consensus mechanisms and their applications in iot: A literature survey”. In: *Algorithms and Architectures for Parallel Processing: 20th International Conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part III* 20. Springer. 2020, pp. 564–579.
- [20] Sikho Luzipo and Aurna Gerber. “A Systematic Literature Review of Blockchain Consensus Protocols”. In: *Responsible AI and Analytics for an Ethical and Inclusive Digitized Society: 20th IFIP WG 6.11 Conference on e-Business, e-Services and e-Society, I3E 2021, Galway, Ireland, September 1–3, 2021, Proceedings* 20. Springer. 2021, pp. 580–595.
- [21] Alankrita Aggarwal, Shivani Gaba, and Mamta Mittal. “A comparative investigation of consensus algorithms in collaboration with IoT and blockchain”. In: *Transforming cybersecurity solutions using blockchain*. Springer, 2021, pp. 115–140.

APPENDIX A

Environment

A.1 Development Environment Setup

A.1.1 Set Up Ubuntu

I first tried installing ubuntu over Vmware in windows. But there issues with space etc.
So then I installed it alongside Windows.

A.1.2 Setting Up Necessary Tools

Then I needed to install Visual Studio Code and the desired libraries for Golang, Delve the debugger and other libraries like git etc. Install all the libraries by typing these commands in Terminal. Terminal of VS Code can also be used.

Install curl

```
sudo snap install curl
```

Removing Go

This step is necessary in order to ensure correct installation of go.

APPENDIX A: ENVIRONMENT

```
sudo apt-get remove golang-go
sudo apt-get remove --auto-remove golang-go
sudo rm -rvf /user/local/go
go version
```

Installing Go

```
sudo apt update && apt upgrade -y
curl -LO https://go.dev/dl/go1.VERSION.linux-amd64.tar.gz
sudo tar -C /usr/local -xzf go1.VERSION.linux-amd64.tar.gz
gedit ~/.bashrc and paste the following lines
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
Then use the following line to apply changes to the env variable
source ~/.bashrc
Go version can be checked by typing the following commands
go version
go env
```

The version command will produce the following results.

```
o version go1.20.4 linux/amd64
```

The env command will produce the following result

```
GO111MODULE=""
GOARCH="amd64"
GOBIN=""
GOCACHE="/home/home/.cache/go-build"
GOENV="/home/home/.config/go/env"
GOEXE=""
```

APPENDIX A: ENVIRONMENT

```
GOEXPERIMENT=""
GOFLAGS=""
GOHOSTARCH="amd64"
GOHOSTOS="linux"
GOINSECURE=""
GOMODCACHE="/home/home/go/pkg/mod"
GONOPROXY=""
GONOSUMDB=""
GOOS="linux"
GOPATH="/home/home/go"
GOPRIVATE=""
GOPROXY="https://proxy.golang.org,direct"
GOROOT="/usr/local/go"
GOSUMDB="sum.golang.org"
GOTMPDIR=""
GOTOOLDIR="/usr/local/go/pkg/tool/linux_amd64"
GOVCS=""
GOVERSION="go1.20.4"
GCCGO="gccgo"
GOAMD64="v1"
AR="ar"
CC="gcc"
CXX="g++"
CGO_ENABLED="1"
GOMOD="/home/home/go-ethereum/go.mod"
GOWORK=""
CGO_CFLAGS="-O2 -g"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-O2 -g"
CGO_FFLAGS="-O2 -g"
```

APPENDIX A: ENVIRONMENT

```
CGO_LDFLAGS="-O2 -g"
```

```
PKG_CONFIG="pkg-config"
```

```
GOGCCFLAGS="-fPIC -m64 -pthread -Wl,--no-gc-sections -fmessage-length=0 -fdebug-prefix-m
```

Installing Git

```
sudo apt-get install git-all
```

Installing Delve Debugger

```
go get github.com/go-delve/delve/cmd/dlv
```

Compiling The Code

The code can be compiled by using the following command.

```
go build -o ~/go/bin/ -v -gcflags="all=-N -l" ./...
```

A.2 Ethereum Environment

A.2.1 Create Directory For the Nodes

In the home directory or anywhere you like create a directory for storing the nodes of the private network.

```
mkdir Nodes
```

```
cd Nodes
```

A.2.2 Create Nodes

Create 4 nodes inside this directory and create accounts.

APPENDIX A: ENVIRONMENT

```
geth --datadir ~/Nodes/node1 account new
geth --datadir ~/Nodes/node2 account new
geth --datadir ~/Nodes/node3 account new
geth --datadir ~/Nodes/node4 account new
```

Save the addresses of these accounts as we'll need them later.

A.2.3 Creating Password Files for Accounts

I had one password for all the files so I created a single file called password.sec and accessed it.

A.2.4 Create Genesis File

Genesis file can also be created using puppeth but since it has become obsolete here's a basic genesis file created manually.

```
{
"config": {
  "chainId": 15,
  "homesteadBlock": 0,
  "eip150Block": 0,
  "eip155Block": 0,
  "eip158Block": 0,
  "byzantiumBlock": 0,
  "constantinopleBlock": 0,
  "petersburgBlock": 0,
  "pledge": {
    "period": 5,
    "epoch": 30000
  }
}
```


A.2.6 Creating Script For Running The Nodes

All the nodes should have different ports. And Miner nodes will need the account for setting etherbase. Create a new file with a .sh extension and paste the following code inside it. This file should have the execute permissions.

Miner Node 1

```
geth --identity "node1" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node1/"
--nodiscover --ws --ws.port=8011 --ws.origins="*" --ws.addr "127.0.0.1"
--http --http.corsdomain "*" --http.port 8041 --http.api
"db,eth,net,personal,admin,miner,web3,pledge" --port 30301 --authrpc.port 8451 --unlock
--password ~/Nodes/password.sec --unlock 0x10C5BD4F90A6e3cB63F93A761A8192f4F12A6B74
--allow-insecure-unlock --mine --miner.etherbase=0x10c5bd4f90a6e3cb63f93a761a8192f4f12a6
--cache 2048 --log.debug --ipcpath "~/Nodes/node1/geth.ipc" console
```

Miner Node 2

```
geth --identity "node2" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node2/"
--nodiscover --ws --ws.port=8012 --ws.origins="*" --ws.addr "127.0.0.1"
--http --http.corsdomain "*" --http.port 8042 --http.api
"db,eth,net,personal,admin,miner,web3,pledge" --port 30302 --authrpc.port 8452
--unlock 0 --password ~/Nodes/password.sec --unlock 0x79b98c69e932f8ffc474e59a24318d9b56
--allow-insecure-unlock --mine --miner.etherbase=0x79b98c69E932f8ffc474e59a24318D9b56007
--cache 2048 --log.debug --ipcpath "~/Nodes/node2/geth.ipc" console
```

Full Node 3

```
geth --identity "node3" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node3"
--nodiscover --ws --ws.port=8013 --ws.origins="*" --ws.addr "127.0.0.1"
--http --http.corsdomain "*" --http.port "8043" --http.api
```


APPENDIX A: ENVIRONMENT

```
"db, eth, net, personal, admin, miner, web3, pledge" --port "30303"  
--authrpc.port 8453 --unlock 0 --password ~/Nodes/node3/password.sec  
--allow-insecure-unlock --cache 2048 --ipcpath "~/Nodes/node3/geth.ipc" --log.debug con
```

Miner Node 4

```
geth --identity "node4" --networkid 52419 --syncmode "full" --datadir "~/Nodes/node4/"  
--nodiscover --ws --ws.port=8014 --ws.origins="*" --ws.addr "127.0.0.1"  
--http --http.corsdomain "*" --http.port 8044 --http.api  
"db, eth, net, personal, admin, miner, web3, pledge" --port 30304 --authrpc.port 8454  
--unlock 0 --password ~/Nodes/password.sec  
--unlock 0xe289d1866c068cd9D7610e955dDd6784feCE1A93 --allow-insecure-unlock  
--mine --miner.etherbase=0xe289d1866c068cd9D7610e955dDd6784feCE1A93  
--cache 2048 --log.debug --ipcpath "~/Nodes/node4/geth.ipc" console
```

A.2.7 Run The Nodes

Now to run the nodes in multiple terminals by navigating into the node directory and running the shell file like

```
./start.sh
```

A.2.8 Forming The Network

In order to connect the nodes to form a network you'll need to add the remaining nodes as peers to the first node by typing the following command in the javascript console of first node.

```
admin.addPeer(ENODEID_OF_THE_NODE)
```

A.2.9 Authorizing Other Nodes

You can authorize second node by typing the following commands. It will need majority of nodes in order for the signer to be approved.

```
pledge.Propose(' ACCOUNT_ADDRESS' , true)
```

A.2.10 Running Delve

After installing Delve it will appear in the left sidebar. To connect it to the running node, click on "Attach To Process" and select the running process by searching for the process. This will show an error as follows.

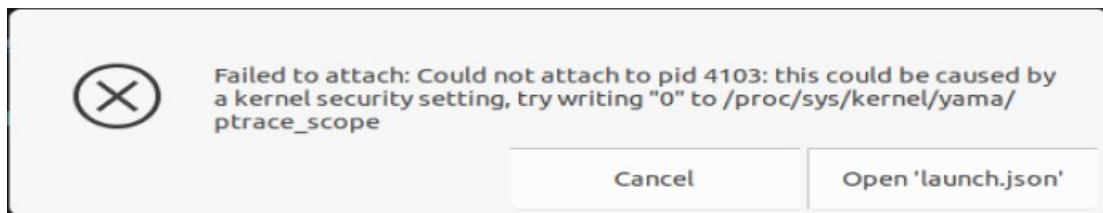


Figure A.4: Delve Error

To resolve this, go to

```
/proc/sys/kernel/yama/ptrace_scope
```

path and write '0' BY Clicking "Save as Sudo".

A.3 Smart Contract

We'll need smart contract to send and receive transaction from the network. For this we write smart contract in Solidity Using RemixIDE. The smart contract can be deployed on the network using the HTTP Port address of the node to which it needs to be deploys.