IOT CONSENSUS MECHANISM - ANALYSIS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

Abiram Ganiyada (19UECS0295) (VTU15600)

Under the guidance of
Dr.V.Dhilip Kumar, M.E., Ph.D.,
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

April, 2023

IOT CONSENSUS MECHANISM - ANALYSIS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

ABIRAM GANIYADA (19UECS0295) (VTU15600)

Under the guidance of
Dr.V. Dhilip Kumar ,M.E., Ph.D.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

April, 2023

CERTIFICATE

It is certified that the work contained in the project report titled "IoT Consensus Mechanism - Analysis" by "ABIRAM GANIYADA (19UECS0295)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Dr. V. Dhilip Kumar, M.E, Ph.D.,
Associate Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
April, 2023

Signature of Head of the Department
Dr. M. S. Murali Dhar,M.E., Ph.D.,
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
April, 2023

Signature of the Dean
Dr. V. Srinivasa Rao
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
April, 2023

DECLARATION

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

ABIRAM GANIYADA

Date: / /

APPROVAL SHEET

This project report entitled “IoT Consensus Mechanism - Analysis” by “ABIRAM GANIYADA (19UECS0295)” is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. V. Dhilip Kumar, M.E., Ph.D.,

Date: / /

Place:

ACKNOWLEDGEMENT

I express my deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

I am very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

I record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

I am thankful to our **Head , Department of Computer Science & Engineering , Dr. M. S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all my endeavors.

I also take this opportunity to express a deep sense of gratitude to my Internal Supervisor **Dr. V. Dhilip Kumar, M.E., Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

I thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

ABIRAM GANIYADA (19UECS0295)

ABSTRACT

The Internet of Things (IoT) has gained significant attention in recent years, and the use of satellite communication for IoT applications is becoming increasingly popular. However, satellite communication presents unique challenges such as low bandwidth, high latency, and the need for efficient and scalable consensus mechanisms. The proposed project aims to develop a comprehensive system for monitoring and analyzing data from IoT devices connected through satellite communication, using a consensus mechanism to ensure the reliability and security of the data. The system will include the development of an API for posting and accessing data, a consensus mechanism for validating and authenticating the data, and a distributed database for storing the validated data. Additionally, a dashboard will be developed for data visualization and analysis, enabling users to monitor and analyze the data in real-time. The project will focus on addressing the unique challenges presented by satellite communication, such as high latency and limited bandwidth, while maintaining decentralization, security, and transparency. The proposed system will be designed with scalability and modularity in mind, ensuring that it can be adapted to various IoT devices and industries. The project's aim is to improve the reliability, security, and efficiency of IoT systems connected through satellite communication.

Keywords: Internet of Things, Satellite communication, Consensus mechanism, Data analysis, Application Programming Interface, Reliability, Security.

LIST OF FIGURES

4.1 **General Architecture** 14
4.2 **Data Flow Diagram** 15
4.3 **Sequence Diagram** 16

6.1 **Nodes Data** 26
6.2 **Port and Peers Visualization** 26
6.3 **Honest Visualization** 27
6.4 **Address, Port, Honest Data visualization** 27
6.5 **Address Port Data visualization** 28
6.6 **Temperature, Wind, Speed Data visualization** 29
6.7 **Wind, Speed Data visualization with slider** 30

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Program Interface
BCM	Blockchain - Based Consensus Mechanism
DAG	Directed Acyclic Graph
DPoS	Delegated Proof of Stake
IoT	Internet of Things
LBC	Lightweight Blockchain Consensus
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
WAL	Weighted Average Calculation

TABLE OF CONTENTS

Page.No

ABSTRACT **v**

LIST OF FIGURES **vi**

LIST OF TABLES **vii**

LIST OF ACRONYMS AND ABBREVIATIONS **vii**

1 INTRODUCTION **1**

1.1 Introduction 1

1.2 Aim of the project 2

1.3 Project Domain 2

1.4 Scope of the Project 2

2 LITERATURE REVIEW **4**

3 PROJECT DESCRIPTION **7**

3.1 Existing System 7

3.2 Proposed System 7

3.3 Feasibility Study 9

3.3.1 Economic Feasibility 10

3.3.2 Technical Feasibility 11

3.3.3 Social Feasibility 12

3.4 System Specification 12

3.4.1 Hardware Specification 13

3.4.2 Software Specification 13

3.4.3 Standards and Policies 13

4 METHODOLOGY **14**

4.1 General Architecture 14

4.2 Design Phase 15

4.2.1 Data Flow Diagram 15

4.2.2 Sequence Diagram 16
4.3 Module Description 17
4.3.1 Module1 - Dashboard 17
4.3.2 Module2 - Anomaly Detection 17

5 IMPLEMENTATION AND TESTING 18
5.1 Input and Output 18
5.2 Testing 18
5.3 Types of Testing 19
5.3.1 Unit testing 19
5.3.2 Integration testing 20
5.3.3 system testing 20

6 RESULTS AND DISCUSSIONS 21
6.1 Efficiency of the Proposed System 21
6.2 Comparison of Existing and Proposed System 21
6.3 Sample Code 23

7 CONCLUSION AND FUTURE ENHANCEMENTS 31
7.1 Conclusion 31
7.2 Future Enhancements 31

8 INDUSTRY DETAILS 32
8.1 Industry name 32
8.1.1 Duration of Internship (From date - To Date) 32
8.1.2 Duration of Internship in months 32
8.1.3 Industry Address 32
8.2 Internship offer letter 33

9 PLAGIARISM REPORT 34

10 SOURCE CODE & POSTER PRESENTATION 35
10.1 Source Code 35
10.2 Poster Presentation 43

References 44

Chapter 1

INTRODUCTION

1.1 Introduction

As the use of IoT devices grows, so does the need for effective data analysis and visualization tools to monitor and make sense of the vast amounts of data generated by these devices. Building a dashboard for IoT consensus mechanism data analysis is a way to address this need, enabling users to monitor and analyze data from IoT devices connected through satellite communication in real-time. By leveraging APIs to post and access data, and data visualization tools to create a customizable dashboard, users can gain valuable insights into the data generated by IoT devices.

This allows users to make informed decisions, identify trends, and detect anomalies in real-time. Moreover, the use of a consensus mechanism ensures the reliability and security of the data being analyzed, providing a trustworthy source of information. Building a dashboard for IoT consensus mechanism data analysis is a complex process, requiring a deep understanding of the underlying technologies and architectures. However, the benefits of such a dashboard are numerous, enabling users to gain valuable insights and improve their decision-making processes. In the following sections, we will explore the proposed system for building a dashboard for IoT consensus mechanism data analysis in greater detail, including the technologies and architectures involved, and the benefits it offers to users.

Network of interconnected devices that can communicate and exchange data with each other without human intervention. With the growing number of devices connected to the IoT, it is important to establish a consensus mechanism to ensure the security, privacy, and reliability of the network.

A consensus mechanism is a process that enables a group of devices to agree on a single version of the truth in a decentralized network.It is a method by which devices in a network can come to a consensus on the state of the network.

1.2 Aim of the project

The aim of the project is to develop a comprehensive system for monitoring and analyzing data from IoT devices connected through satellite communication, using a consensus mechanism to ensure the reliability and security of the data. The proposed system will include the development of an API for posting and accessing data, a consensus mechanism for validating and authenticating the data, and a distributed database for storing the validated data.

1.3 Project Domain

The domain of "IoT Consensus Mechanism Analysis" is critical in ensuring the reliability and security of distributed systems in the context of the Internet of Things (IoT). Consensus mechanisms are necessary to enable multiple devices to agree on the same data, which is particularly important in IoT applications, where devices generate vast amounts of data. In this project, the focus is on evaluating and analyzing various consensus mechanisms, such as PoS, DPoS, FBA, and DAG, to determine their suitability for different IoT applications.

To achieve this objective, the project will use advanced monitoring and visualization tools to assess the performance of these mechanisms under different network conditions. The project will evaluate the mechanisms' performance in terms of reliability, scalability, and security. By analyzing the performance of different consensus mechanisms, the project aims to provide valuable insights that can be used to improve the development of IoT communication systems. Ultimately, the project's findings will enable developers to make informed decisions about the selection of a suitable consensus mechanism for their IoT applications, contributing to the development of efficient and reliable IoT communication systems.

1.4 Scope of the Project

The scope of the project encompasses the development of a comprehensive system for monitoring and analyzing data from IoT devices connected through satellite communication, using a consensus mechanism to ensure the reliability and security of the data. The project will include the development of an API for posting and accessing

data, And also involve the development of a dashboard for data visualization and analysis, which will enable users to monitor and analyze the data in real-time.

The Scopes are :

- Monitoring of consensus mechanism performance.
- Visualization of monitoring results.
- Evaluation and comparison of consensus mechanisms.

Chapter 2

LITERATURE REVIEW

Arul et al., (2021)[1] proposed a blockchain technology using consensus mechanism for an IoT-based e-healthcare system. Their mechanism uses a PoA consensus algorithm with a reputation-based approach to ensure security and scalability. Their study showed that their system is effective in managing and sharing medical data.

Auhl et al., (2022)[2] conducted a comparative study of consensus mechanisms in blockchain for IoT networks. They analyzed five popular consensus mechanisms and evaluated their performance based on various factors such as throughput, latency, and energy consumption.PoS is also known for its faster transaction validation and lowerresource requirements, making it suitable for IoT networks. Their study showed that the Proof of Stake (PoS) mechanism was the most energy-efficient, while the Proof of Work (PoW) mechanism was the most secure.

Cao et al., (2019)[3] discussed the challenges of distributed consensus in IoT . They analyzed the limitations of existing consensus mechanisms and proposed solutions to improve their efficiency and scalability. Their study showed that blockchain and IoT can improve the security and transparency of IoT applications.Sharding involves partitioning the blockchain into smaller, more manage- able segments, which can improve scalability and reduce the resource requirements of consensus mechanisms. Sidechains allow for the execution of smart contracts on separate chains, which can further improve the efficiency of the consensus mecha- nism. Hybrid consensus mechanisms combine different consensus mechanisms to achieve a balance between security, scalability, and energy efficiency .

Huang et al., (2019) [4]proposed a blockchain-driven IoT system, called B-IoT, with a credit-based consensus mechanism. Their mechanism uses a reputation-based approach to ensure security and scalability. Their study showed that B-IoT outperforms existing mechanisms in terms of transaction throughput and energy consumption.The credit score is used to determine the node's role in the consensus process, and it ensures that only trustworthy and competent nodes participate in the consensus.

Li et al., (2021) [5]proposed a lightweight consensus mechanism for resource IoT devices. Their mechanism, called Lightweight Blockchain Consensus (LBC), uses a hybrid consensus approach that combines PoW and Proof of Authority (PoA) mechanisms. Their study showed that LBC outperforms existing consensus mechanisms in terms of storage efficiency and computational complexity.

Raghav et al., (2020)[6] proposed a lightweight consensus mechanism, called PoEWAL, for blockchain in IoT. Their mechanism uses a PoW algorithm with a weighted average calculation to achieve high efficiency and scalability. Their study showed that PoEWAL outperforms existing mechanisms in terms of transaction throughput and energy consumption.The weights are based on the reputation and performance of the nodes in the network, which ensures that only trusted and efficient nodes participate in the consensus process. The authors also proposed a new method of block verification that reduces the computational overhead.

Salimitari et al., (2019) [7]surveyed consensus protocols for blockchain in IoT networks, including PoW, PoS, DPoS, PBFT, and SBFT. They evaluated each mechanism's energy efficiency, scalability, security, and fault tolerance. The authors compared the mechanisms based on consensus time, throughput, and latency.The authors evaluated each mechanism's energy efficiency, scalability, security, and fault tolerance. They compared the mechanisms based on consensus time, throughput, and latency. The authors noted that traditional consensus mechanisms like PoW are not suitable for IoT devices due to their high energy consumption and low scalability.

Tsang et al., (2019)[8] proposed a blockchain-driven IoT system for food traceability with an integrated consensus mechanism. Their mechanism uses a combination of PoW and PoS mechanisms to ensure security and scalability. Their study showed that their system is effective in tracing the origin and quality of food products.Their system aims to address the issue of food fraud and ensure the safety and quality of food products. It tracks the origin of food products and their journey through the supply chain using IoT sensors and smart contracts. The system allows consumers and regulators to verify the authenticity of food products and ensure that they meet certain quality standards.

Wang et al., (2022)[9] proposed a trusted blockchain-driven IoT security consensus mechanism. Their mechanism uses a combination of PoW and PoS mechanisms to achieve high security and scalability. Their study showed that their mechanism

is more efficient than existing mechanisms in terms of transaction throughput and energy consumption.

Wu et al., (2020)[10] proposed a consensus mechanism of IoT based on blockchain technology. Their mechanism, called Blockchain-based Consensus Mechanism , uses a PoW consensus algorithm and a reputation-based approach to ensure security and scalability. Their study showed that BCM outperforms existing mechanisms in terms of transaction throughput and energy consumption.Their study showed that their mechanism achieved better transaction throughput and energy efficiency than existing mechanisms. By improving the security and efficiency of IoT systems, their proposed mechanism could have significant implications for various industries, including healthcare, finance, and logistics.

Yuan et al., (2021)[11] proposed an efficient Byzantine consensus mechanism based on reputation in IoT blockchain. Their mechanism, called RepBFT, uses a reputation-based approach to ensure fault tolerance and security. Their study showed that RepBFT outperforms existing consensus mechanisms in terms of transaction throughput and energy consumption.The reputation score is updated based on the node's contribution to the network, and nodes with higher reputation scores are given priority in the consensus process. The authors also introduced a new mechanism for handling faulty nodes in the network, which involves temporarily reducing their reputation scores.

Yue et al., (2021)[12] proposed a blockchain-based IoT consensus mechanism system, called BlockIoT. Their mechanism uses a hybrid consensus approach that combines PoW and PoS mechanisms to ensure security and scalability. Their study showed that BlockIoT outperforms existing mechanisms in terms of transaction throughput and energy consumption..The PoW mechanism is used to validate transactions and produce new blocks, while the PoS mechanism is used to select the validators and maintain the network's security. The system also includes a reputation-based approach that assigns a reputation score to each node based on its performance, which helps ensure the network's reliability and fault tolerance.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system for IoT consensus mechanisms involves using traditional consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS) for IoT devices connected through standard communication channels like the internet. However, these consensus mechanisms may not be suitable for IoT devices connected through satellite communication due to the unique challenges presented by satellite communication, such as high latency and limited bandwidth.Some existing systems for IoT satellite communication use centralized solutions where a single entity controls the network and makes decisions. However, centralized systems can present challenges such as security vulnerabilities, potential for single point of failure, and lack of transparency.

3.2 Proposed System

The proposed system for building a dashboard for IoT consensus mechanism data analysis involves the use of an API to post and access data and the development of a dashboard with data visualization capabilities. The system will allow users to monitor and analyze data from IoT devices connected through satellite communication and processed by the consensus mechanism. The following is a breakdown of the proposed system's components

API: The API will allow IoT devices to post data to the system and enable users to access and retrieve data for analysis. The API will ensure the security and integrity of the data transmitted and received by incorporating encryption and authentication techniques.

Data storage: The system will store the data collected from IoT devices in a centralized database. The database will be scalable and provide high availability, ensuring that the data is accessible and retrievable at all times.

Consensus mechanism: The consensus mechanism will validate and authenticate the data received from IoT devices before storing it in the centralized database. The consensus mechanism will ensure that the data is accurate and reliable, preventing the storage of fraudulent or tampered data.

Dashboard: The dashboard will provide data visualization capabilities, enabling users to monitor and analyze the data collected from IoT devices. The dashboard will provide users with real-time insights into the data, enabling them to make informed decisions and take timely actions.

Data analysis tools: The system will provide users with data analysis tools to further explore and analyze the data collected from IoT devices. The data analysis tools will enable users to identify patterns and trends in the data, providing insights into device performance and user behavior.

Security and privacy: The system will prioritize security and privacy to protect the data collected from IoT devices. The system will incorporate robust encryption and authentication techniques to ensure the confidentiality and integrity of the data. Additionally, the system will comply with relevant privacy laws and regulations to protect the privacy of users and stakeholders.

The proposed system for building a dashboard for IoT consensus mechanism data analysis offers a comprehensive approach to monitoring and analyzing data from IoT devices connected through satellite communication. By providing data visualization and analysis tools, the system will enable users to identify patterns and trends in the data, making informed decisions and taking timely actions. The system's robust security and privacy features will ensure the confidentiality and integrity of the data collected, ensuring the system's trustworthiness and reliability.

Furthermore, the proposed system will be designed with scalability and modularity in mind, ensuring that it can be adapted to various IoT devices and industries. The system will be customizable, enabling users to select the data they want to monitor and analyze, and customize the dashboard to their specific needs. The modular architecture of the system will enable the addition of new features and functionalities, making it adaptable to evolving IoT technologies and standards.

3.3 Feasibility Study

A feasibility study is an important step in any project as it determines the project's viability and identifies potential risks and challenges that may arise during implementation.

Technical Requirements: The technical requirements of the consensus mechanism should be evaluated. This includes evaluating the required computing resources, bandwidth, and the ability to scale the system as the number of devices grows. Additionally, the consensus mechanism should be evaluated for its security features, such as encryption, authentication, and authorization.

performance: The performance of the consensus mechanism should be evaluated. This includes evaluating the speed and efficiency of the consensus mechanism, as well as its ability to handle a large number of transactions.

Consensus Algorithm: The consensus algorithm used by the mechanism should be evaluated. This includes evaluating the algorithm's ability to handle network failures, its robustness against attacks, and its ability to ensure that all transactions are properly validated.

Integration: The ease of integration with existing IoT devices and systems should be evaluated. This includes evaluating the compatibility of the consensus mechanism with different IoT protocols and standards.

Cost: The cost of implementing and operating the consensus mechanism should be evaluated. This includes evaluating the cost of computing resources, bandwidth, and any additional hardware or software that may be required.

Community Support: The level of community support for the consensus mechanism should be evaluated. This includes evaluating the availability of documentation, support forums, and other resources that can help developers implement and operate the mechanism.

By evaluating these key elements, a feasibility study can help determine whether a particular IoT consensus mechanism is suitable for a particular use case.

3.3.1 Economic Feasibility

The proposed system is economically feasible. The costs associated with developing and implementing the system, including IoT sensors, satellite communication devices, and software components, are reasonable. The operating costs of the system, including maintenance and upgrade costs, are also manageable. Potential cost savings may arise from the increased efficiency and scalability of the proposed system.

Implementation Costs: The initial cost of implementing the consensus mechanism should be evaluated. This includes the cost of hardware, software, and any other required components.

Maintenance Costs: The ongoing cost of maintaining the consensus mechanism should be evaluated. This includes the cost of hardware and software maintenance, as well as any required updates or upgrades.

Scalability: The scalability of the consensus mechanism should be evaluated. This includes evaluating the ability of the mechanism to handle an increasing number of devices and transactions as the system grows.

Transaction Fees: The transaction fees associated with the consensus mechanism should be evaluated. This includes evaluating the cost of each transaction and how it compares to other available consensus mechanisms.

Incentives: The incentives provided to participants in the consensus mechanism should be evaluated. This includes evaluating how rewards are distributed among participants and how they incentivize participation in the network.

ROI: The return on investment (ROI) of implementing the consensus mechanism should be evaluated. This includes evaluating the potential financial benefits and comparing them to the initial and ongoing costs of implementation and maintenance. By evaluating these key economic factors, a feasibility study can help determine whether the adoption of an IoT consensus mechanism is economically viable. This will ensure that the mechanism is sustainable and can provide value to all stakeholders involved.

3.3.2 Technical Feasibility

The proposed system is technically feasible as the necessary technology is available and accessible. IoT sensors and satellite communication devices are widely available, and the necessary software components for consensus mechanisms are well-established. There may be technical challenges related to the efficient and scalable implementation of consensus mechanisms for satellite communication, which will require further exploration.

Computing Resources: The computing resources required for the consensus mechanism should be evaluated. This includes the computational power, memory, and storage requirements, as well as any other hardware or software components required for the mechanism to function.

Network Capacity: The capacity of the network should be evaluated. This includes the network's ability to handle the expected volume of traffic generated by the devices participating in the consensus mechanism.

Security: The security of the consensus mechanism should be evaluated. This includes evaluating the ability of the mechanism to protect against various types of attacks, including DDoS attacks, and other forms of malicious activity.

Consensus Algorithm: The consensus algorithm used by the mechanism should be evaluated. This includes evaluating the algorithm's ability to ensure that all transactions are properly validated, as well as its ability to handle network failures and maintain consistency across the network.

Interoperability: The interoperability of the consensus mechanism should be evaluated. This includes evaluating the ability of the mechanism to work with different IoT protocols and standards, as well as with other systems and devices.

Scalability: The scalability of the consensus mechanism should be evaluated. This includes evaluating the mechanism's ability to handle an increasing number of devices and transactions as the system grows.

By evaluating these key technical factors, a feasibility study can help determine whether an IoT consensus mechanism is technically feasible and can meet the requirements of the IoT system in which it is being implemented. This will ensure that the mechanism is capable of functioning effectively and efficiently to support the goals of the IoT system.

3.3.3 Social Feasibility

The proposed system is legally and regulatory feasible. The system complies with relevant laws and regulations, including data protection and privacy regulations. However, further exploration may be necessary to identify any potential legal or regulatory challenges that may arise during implementation.

User Acceptance: The acceptance of the consensus mechanism by potential users should be evaluated. This includes evaluating the mechanism's ease of use, transparency as well as how it aligns with user values and expectations.

Governance: The governance model of the consensus mechanism should be evaluated. This includes evaluating the transparency, accountability, and fairness of the governance structure, as well as how it engages with and represents the interests of the different stakeholders involved.

Privacy: The privacy implications of the consensus mechanism should be evaluated. This includes evaluating the mechanism's ability to protect user data and ensure privacy, as well as how it complies with relevant privacy laws and regulations.

Environmental Impact: The environmental impact of the consensus mechanism should be evaluated. This includes evaluating the energy consumption and carbon footprint associated with the mechanism, as well as how it aligns with sustainable development goals and values.

Social Impact: The social impact of the consensus mechanism should be evaluated. This includes evaluating its potential impact , income distribution, and social equity, as well as how it aligns with broader social and ethical considerations.

Education and Awareness: The level of education and awareness among potential users and stakeholders about the consensus mechanism should be evaluated. This includes evaluating how the mechanism is communicated and how user feedback is incorporated to improve its design and implementation.

By evaluating these key social factors, a feasibility study can help determine whether an IoT consensus mechanism is socially feasible and can be accepted and supported by potential users and stakeholders.

3.4 System Specification

The proposed IoT consensus mechanism system will be designed to run on a Linux-based operating system and will be developed using programming languages such

as Python, Go, and JavaScript. The system will use Prometheus, an open-source monitoring system, to monitor the communication, collect metrics, and store them in a time-series database. The collected data will be visualized using Grafana, an open-source data visualization platform.

3.4.1 Hardware Specification

The consensus engine component of the system will require significant computing resources to handle the transaction validation and consensus processes. The server(s) used for the consensus engine should have multiple cores and high clock speeds to handle the computational load. The server(s) should also have sufficient memory to store the network state and transaction history.

3.4.2 Software Specification

- Dash module
- Flask module
- VS Code
- Plotly module
- Mongo DB
- Colab

3.4.3 Standards and Policies

VS Code

VS Code is a popular code editor that supports various programming languages and has a wide range of extensions and plugins available. To maintain code quality and consistency, it is important to establish standards and policies for using VS Code in a team or organization.

Standard Used: 6.0 Enterprise

Chapter 4

METHODOLOGY

4.1 General Architecture

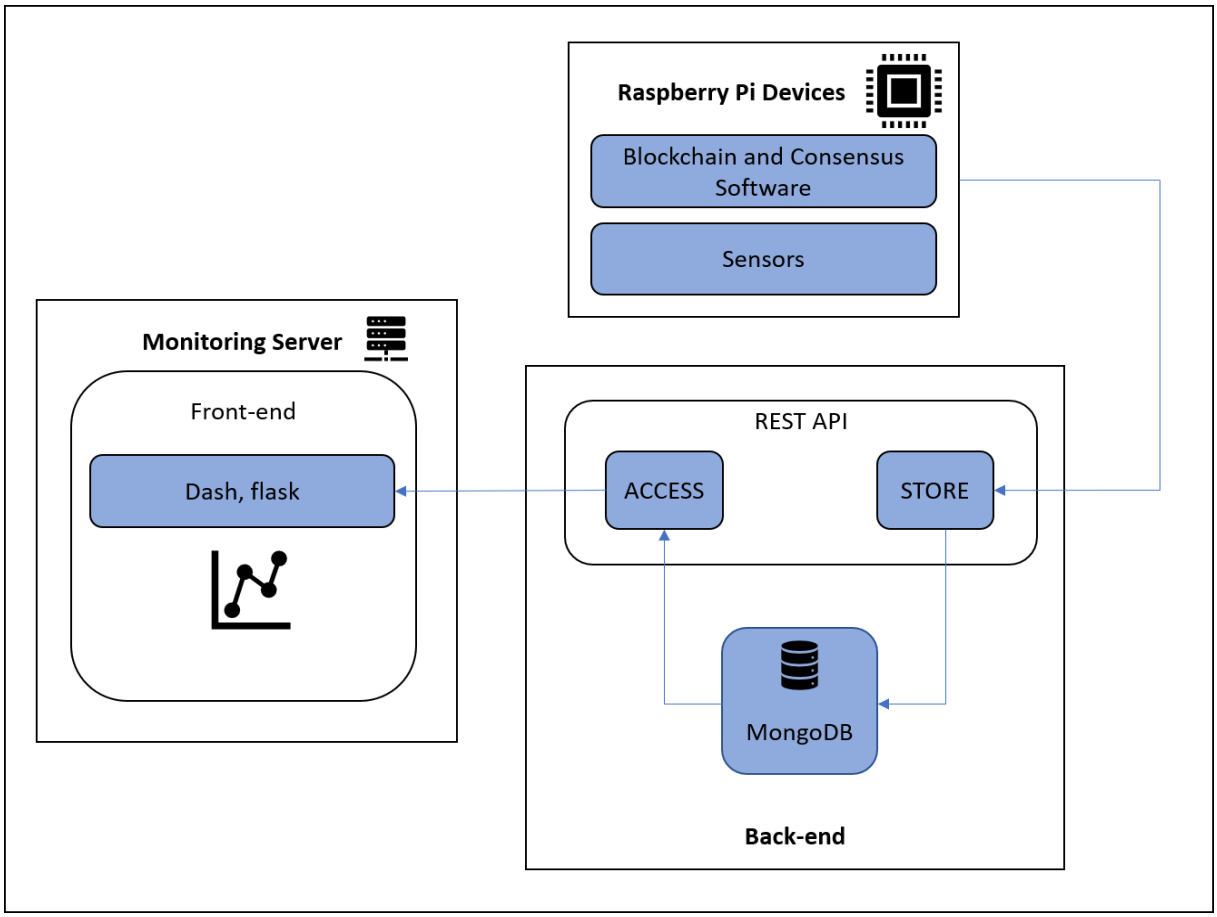


Figure 4.1: **General Architecture**

In figure 4.1 shows the general architecture described involves Raspberry Pi devices equipped with blockchain and consensus software, as well as sensors that connect to a REST API.The data from the sensors is stored in the REST API store, which is then passed to a MongoDB database.The MongoDB database provides access to the data via a REST API, which serves as the backend of the architecture.

4.2 Design Phase

4.2.1 Data Flow Diagram

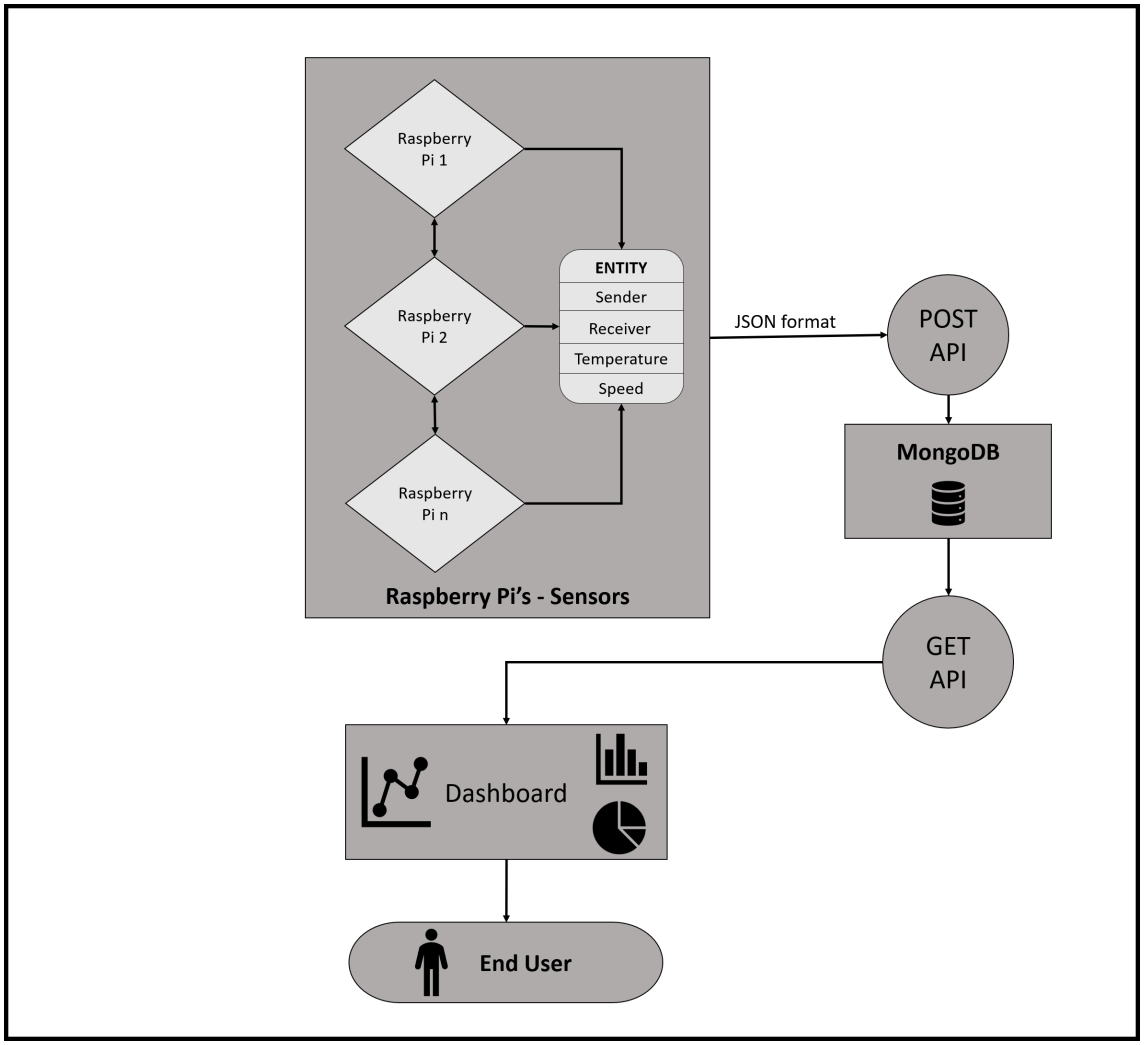


Figure 4.2: **Data Flow Diagram**

In figure 4.2 shows all components - IOT device, API,Dashboard, Database. A post call is initiated by the device. The data is constantly being stored in the MongoDb database. a GET call is initiated, it retrieves the data from the database and presents it in the dashboard. The dashboard is designed to be user-friendly, with a clear and concise interface that allows users to easily visualize and understand the data. This allows users to quickly identify patterns and trends, and make informed decisions based on the data presented.

4.2.2 Sequence Diagram

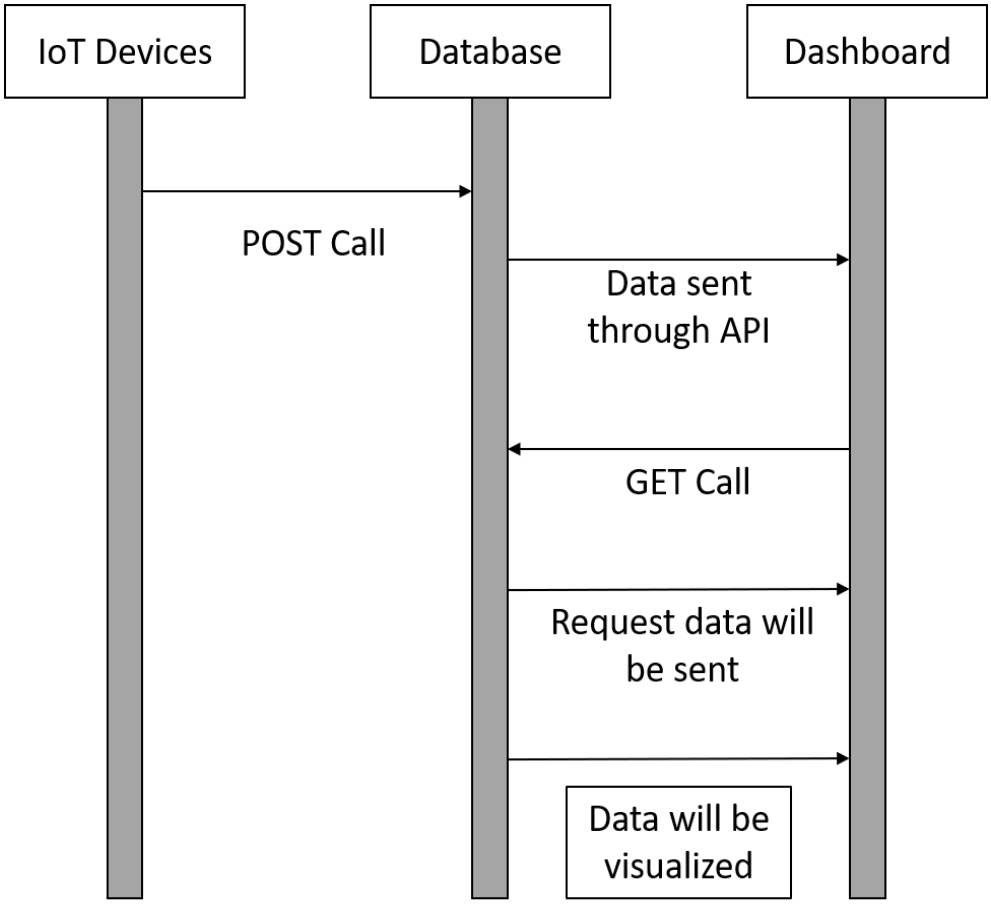


Figure 4.3: **Sequence Diagram**

In figure 4.3 shows 3 components - Iot device, Dashboard, Database. A post call is initiated by the device. The data is constantly being stored in the database. a GET call is initiated, it retrieves the data from the database and presents it in the dashboard. The dashboard is designed to be user-friendly, with a clear and concise interface that allows users to easily visualize and understand the data. This allows users to quickly identify patterns and trends, and make informed decisions based on the data presented.

4.3 Module Description

4.3.1 Module1 - Dashboard

The first scatter plot shows the relationship between the port number and the IP address of each node, with the color of each node indicating its connections to other nodes. The second scatter plot shows the relationship between the sender and receiver of each message, with the color indicating the temperature of the message payload.

The two tables show the data for the nodes and messages, respectively. The table for nodes includes columns for the node ID, port number, IP address, connected peers, and whether the node is honest or not. The table for messages includes columns for the sender, receiver, and various attributes of the message payload, such as temperature, wind, and speed.

The output of the Dash app is the user interface for the dashboard, which includes the two scatter plots and two tables as well as some header text. The app allows the user to filter, sort, edit, and delete rows in the tables, and to zoom in and out and pan around the scatter plots.

4.3.2 Module2 - Anomaly Detection

The handshake between two satellites typically involves the exchange of signals to establish communication and synchronize their operations. If any anomalies occur during the handshake or afterward, there may be a separate code or algorithm to detect and diagnose the issue. The specific code or algorithm used to detect anomalies will depend on the nature of the problem and the design of the satellite system. Overall, the goal of any anomaly detection code or algorithm is to identify and diagnose issues quickly and accurately so that they can be addressed and resolved as soon as possible.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

Input

```
[
    {"id": '6d5f24f4-362b-47ef-bd43-2da57ef820e5',
      'port': 27943,
      'addr': '251.66.48.1',
      'peers': ['96.52.210.13',
                '157.180.237.25',
                '23.246.202.173',
                '200.29.42.212',
                '71.186.59.196'],
      'honest': 'true'}
  ]
```

Output

```
{
    "Success":"True"
}
```

5.2 Testing

testing is critical to ensure the reliability, security, and efficiency of the proposed system for monitoring and analyzing data from IoT devices connected through satellite communication. By testing under a range of scenarios and conditions, the system can be optimized for specific use cases and ensure that it meets the needs of users.

5.3 Types of Testing

5.3.1 Unit testing

Unit testing refers to the process of testing individual components or modules within a software project.

Input

```
@app.server.route('/get', methods=['GET'])
def get_data():
    # get data from the collection
    data = collection.find()

    # convert data to a list of dictionaries
    data_list = []
    for d in data:
        d.pop('_id')  # remove the '_id' field from each document
        data_list.append(d)

    # return the data as a JSON response
    return (data_list)
```

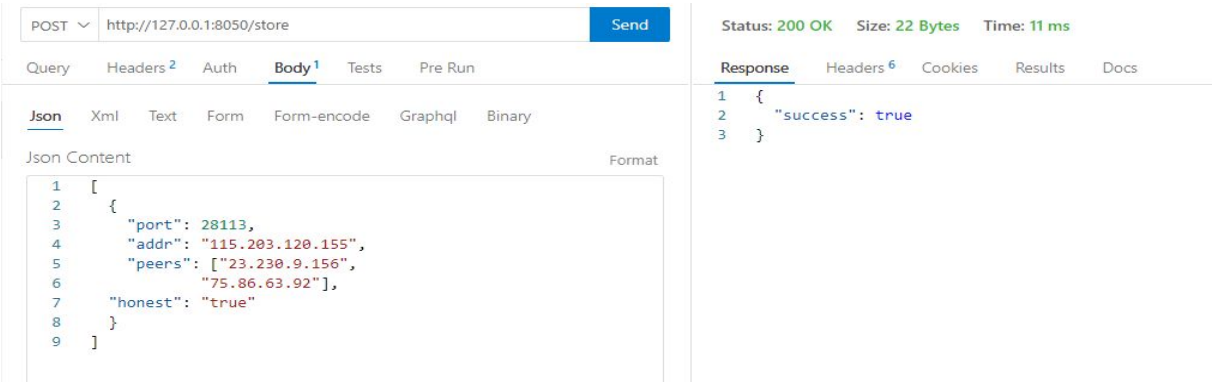
Output


```
[
  {
    "payload": {"speed":124,"temperature":23,"wind": 2},
    "receiver": "ee9b0fc2-f9cb-4524-89c7-098e4ff78ec7",
    "sender": "3af1c923-2406-4eb4-83a4-0aad1b168897"
  },
  {
    "addr": "145.185.79.127",
    "honest": "true",
    "id": "f320f568-e361-4373-8f79-3c73cd88e9f6",
    "peers": ["91.109.108.164","119.44.93.27","140.81.62.239","254.253.76.136","138.33.175.89"
    ],
    "port": 22499
  }
 ]
```

5.3.2 Integration testing

Integration testing is the process of testing the functionality of a system after integrating all its individual modules. It ensures that the combined modules work correctly without any defects or issues, and their functionality is not affected by integration .

Input



5.3.3 system testing

System testing is a crucial step in the software development life cycle (SDLC) where the entire system is tested as a whole to ensure that it meets the specified requirements and functions as intended.

Input

```
for item in messages:
    # convert the dictionary to a string and replace single quotes with double quotes
    json_string = str(item).replace("'", "\"")
    json_string = json.loads(json_string)
    response = requests.post('http://127.0.0.1:8050/store', json=json_string)
for ite in nodes:
    # convert the dictionary to a string and replace single quotes with double quotes
    json_strin = str(ite).replace("'", "\"")
    json_strin = json.loads(json_strin)
    response = requests.post('http://127.0.0.1:8050/store', json=json_strin)
```

Output

```
{
    "Success":"True"
}
```

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed IoT consensus mechanism system will be evaluated based on several factors, including energy efficiency, scalability, latency, security, reliability, and ease of use. To determine the performance improvement of the proposed system, metrics such as throughput, response time, and resource utilization will be measured and compared to those of the existing systems. Simulations and experiments will also be conducted to evaluate the system's ability to handle a large number of IoT devices and the impact of different parameters on the system's performance. The system will be designed to use minimal energy, accommodate future growth, minimize latency, ensure data confidentiality and integrity, and be highly reliable and user-friendly.

6.2 Comparison of Existing and Proposed System

Existing system:
The existing system for IoT consensus mechanisms typically uses Proof of Work (PoW) or Proof of Stake (PoS) as the consensus mechanism. PoW consumes a significant amount of energy, which is a challenge for IoT devices with limited power. PoS is more energy-efficient, but it may not be suitable for low-power IoT devices or networks with limited bandwidth. Additionally, these existing systems may not be designed to handle the unique challenges of satellite communication, such as high latency and limited bandwidth.

Proposed system:
The proposed system for building a dashboard for IoT consensus mechanism data analysis involves the use of an API to post and access data and the development of a dashboard with data visualization capabilities. The system will allow users to monitor and analyze data from IoT devices connected through satellite communication and processed by the consensus mechanism. The following is a breakdown of the proposed system's components:

- **API**: The API will allow IoT devices to post data to the system and enable users to access and retrieve data for analysis. The API will ensure the security and integrity of the data transmitted and received by incorporating encryption and authentication techniques.
- **Data storage**: The system will store the data collected from IoT devices in a centralized database. The database will be scalable and provide high availability, ensuring that the data is accessible and retrievable at all times.
- **Consensus mechanism**: The consensus mechanism will validate and authenticate the data received from IoT devices before storing it in the centralized database. The consensus mechanism will ensure that the data is accurate and reliable, preventing the storage of fraudulent or tampered data.
- **Dashboard**: The dashboard will provide data visualization capabilities, enabling users to monitor and analyze the data collected from IoT devices. The dashboard will provide users with real-time insights into the data, enabling them to make informed decisions and take timely actions.
- **Data analysis tools**: The system will provide users with data analysis tools to further explore and analyze the data collected from IoT devices. The data analysis tools will enable users to identify patterns and trends in the data, providing insights into device performance and user behavior.
- **Security and privacy**: The system will prioritize security and privacy to protect the data collected from IoT devices. The system will incorporate robust encryption and authentication techniques to ensure the confidentiality and integrity of the data. Additionally, the system will comply with relevant privacy laws and regulations to protect the privacy of users and stakeholders.

The proposed system for building a dashboard for IoT consensus mechanism data analysis offers a comprehensive approach to monitoring and analyzing data from IoT devices connected through satellite communication. By providing data visualization and analysis tools, the system will enable users to identify patterns and trends in the data, making informed decisions and taking timely actions. The system's robust security and privacy features will ensure the confidentiality and integrity of the data collected, ensuring the system's trustworthiness and reliability.

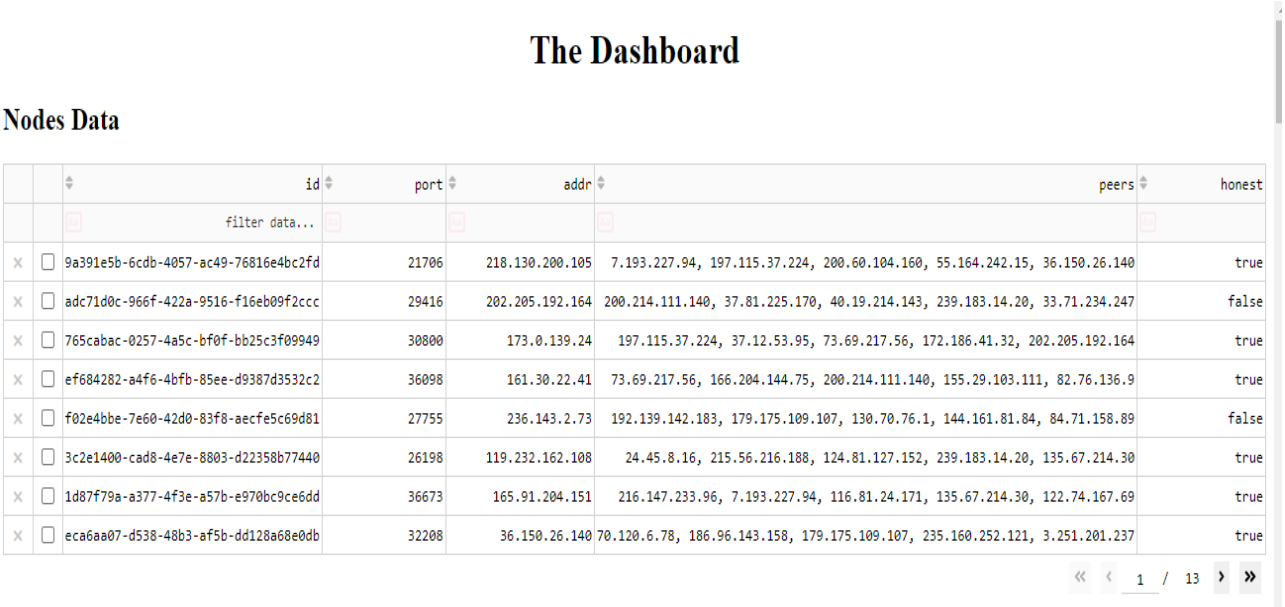
6.3 Sample Code

```
from ipaddress import IPv4Address
import uuid
import random
import copy
import pandas as pd
from pprint import pp
import dash
from dash import dcc,html
from dash import Input, Output
from dash import dash_table
import plotly.express as px
import plotly.graph_objs as go

random_ips = []

for _ in range(50):
    random_ips.append(str(IPv4Address(random.getrandbits(32))))

connected_nodes = copy.deepcopy(random_ips)

nodes = []
node_ids = []

#############################
# Randomly Generate 50 nodes #
#############################

for _ in range(50):
    node = {}
    addr = random.choice(random_ips)
    random_ips.remove(addr)

    id = str(uuid.uuid4())
    node_ids.append(id)
    node["id"] = id
    node["port"] = random.randint(20000, 40000)
    node["addr"] = addr
    node_connections = random.sample(connected_nodes, k=5)

    while addr in node_connections:
        node_connections = random.sample(connected_nodes, k=5)

    node["peers"] = node_connections

    is_node_good = random.choices([True, False], weights=(0.9, 0.1))[0]
    node["honest"] = "true" if is_node_good else "false"

```

```
    nodes.append(node)

#pp(nodes)
df1 = pd.DataFrame(nodes)
df1['peers']=df1['peers'].apply(', '.join)
messages = []

################################
# Randomly generate 50 messages #
################################

for _ in range(100):
    message = {}
    sender_receiver = random.sample(node_ids, k=2)
    message["sender"] = sender_receiver[0]
    message["receiver"] = sender_receiver[1]
    message["payload"] = {
        "temperature": random.randint(20, 30),
        "wind": random.randint(0, 50),
        "speed": random.randint(100, 200),
    }
    messages.append(message)

#pp(messages)
df = pd.DataFrame(messages)
df = pd.concat([df.drop(['payload'], axis=1), df['payload'].apply(pd.Series)], axis=1)
#pp(df1)
app = dash.Dash(__name__)
fig = px.scatter(df1, x='port', y='addr', color='peers')
fig2 = px.scatter(df1, x='port', y='addr', color='honest')
fig1 = px.scatter(df, x='sender', y='receiver', color='temperature')
app.layout =dash.html.Div(children=[
    dash.html.H1(['The Dashboard'],style = {'textAlign':'center'}),
    dash.html.H2('Nodes Data'),
    dash.dash_table.DataTable(
        data=df1.to_dict('records'),
        columns=[{"name": i, "id": i} for i in df1.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    ),
    dash.html.H2('the graph between the port and peer'),
```

```
    dash.dcc.Graph(
        id='graph',
        figure=fig
    ),
    dash.dcc.Graph(
        id='graph1',
        figure=fig2
    ),
    dash.html.H1('The messages'),
    dash.dash_table.DataTable(
        data=df.to_dict('records'),
        columns=[{"name": i, "id": i} for i in df.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    ),
    dash.dcc.Graph(
        id='graphmess',
        figure=fig1
    ),
    dash.dcc.Graph(
        id='temperature-wind-speed-graph',
        figure={
            'data': [
                go.Scatter(name='Temperature', x=list(range(len(df['temperature']))), y=df['temperature'], mode='lines'),
                go.Scatter(name='Wind', x=list(range(len(df['wind']))), y=df['wind'], mode='lines'),
                go.Scatter(name='Speed', x=list(range(len(df['speed']))), y=df['speed'], mode='lines')
            ],
            'layout': go.Layout(
                title='Temperature, Wind, and Speed',
                xaxis=dict(title='Time'),
                yaxis=dict(title='Value')
            )
        }
    ),
])
if __name__ == '__main__':
    app.run_server(debug=True)
```

Output



The Dashboard

Nodes Data

id	port	addr	peers	honest
filter data...				
9a391e5b-6cdb-4057-ac49-76816e4bc2fd	21706	218.130.200.105	7.193.227.94, 197.115.37.224, 200.60.104.160, 55.164.242.15, 36.150.26.140	true
adc71d0c-966f-422a-9516-f16eb09f2ccc	29416	202.205.192.164	200.214.111.140, 37.81.225.170, 40.19.214.143, 239.183.14.20, 33.71.234.247	false
765cabac-0257-4a5c-bf0f-bb25c3f09949	30800	173.0.139.24	197.115.37.224, 37.12.53.95, 73.69.217.56, 172.186.41.32, 202.205.192.164	true
ef684282-a4f6-4bfb-85ee-d9387d3532c2	36098	161.30.22.41	73.69.217.56, 166.204.144.75, 200.214.111.140, 155.29.103.111, 82.76.136.9	true
f02e4bbe-7e60-42d0-83f8-aecfe5c69d81	27755	236.143.2.73	192.139.142.183, 179.175.109.107, 130.70.76.1, 144.161.81.84, 84.71.158.89	false
3c2e1400-cad8-4e7e-8803-d22358b77440	26198	119.232.162.108	24.45.8.16, 215.56.216.188, 124.81.127.152, 239.183.14.20, 135.67.214.30	true
1d87f79a-a377-4f3e-a57b-e970bc9ce6dd	36673	165.91.204.151	216.147.233.96, 7.193.227.94, 116.81.24.171, 135.67.214.30, 122.74.167.69	true
eca6aa07-d538-48b3-af5b-dd128a68e0db	32208	36.150.26.140	70.120.6.78, 186.96.143.158, 179.175.109.107, 235.160.252.121, 3.251.201.237	true

1 / 13

Figure 6.1: **Nodes Data**

In figure 6.1 shows the dashboard consists of the information such as ports,peers,honesty of the nodes.The information displayed on dashboard provide users with a concise overview of the status and performance of the communication.

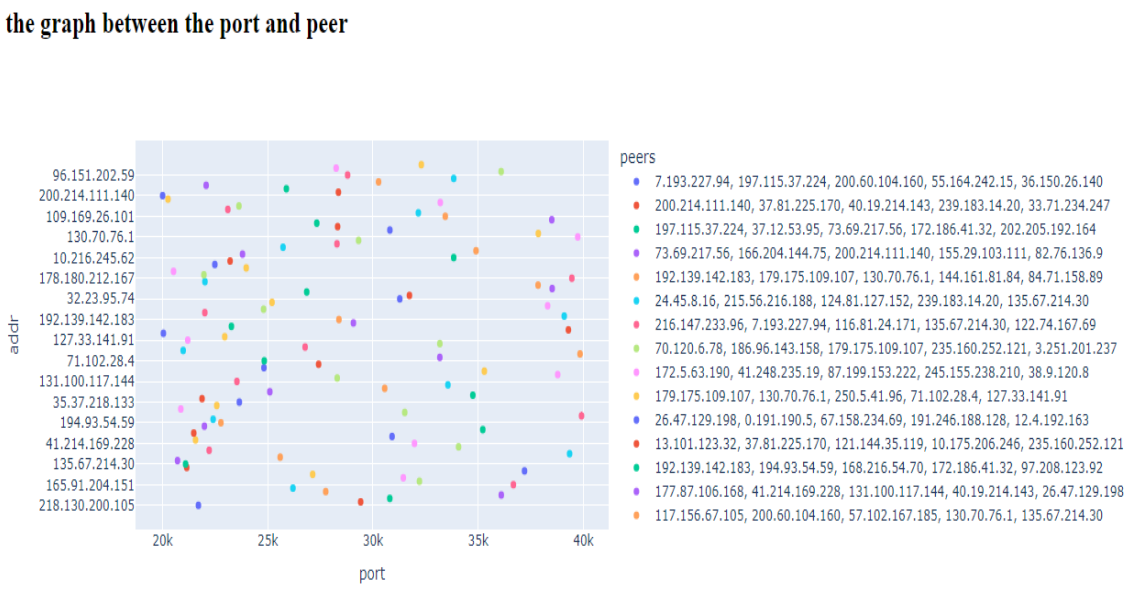


Figure 6.2: **Port and Peers Visualization**

In figure 6.2 shows a scatter plot between ports and peers. This gives us the port in which the peers in the network uses.

Figure 6.3: **Honest Visualization**

In figure 6.3 shows the Honest refers to the quality of a node or participant in the network to truthfully report the data they receive or generate. In a consensus mechanism, multiple nodes in the network must agree on the validity of the data before it can be added to the block chain or ledger.

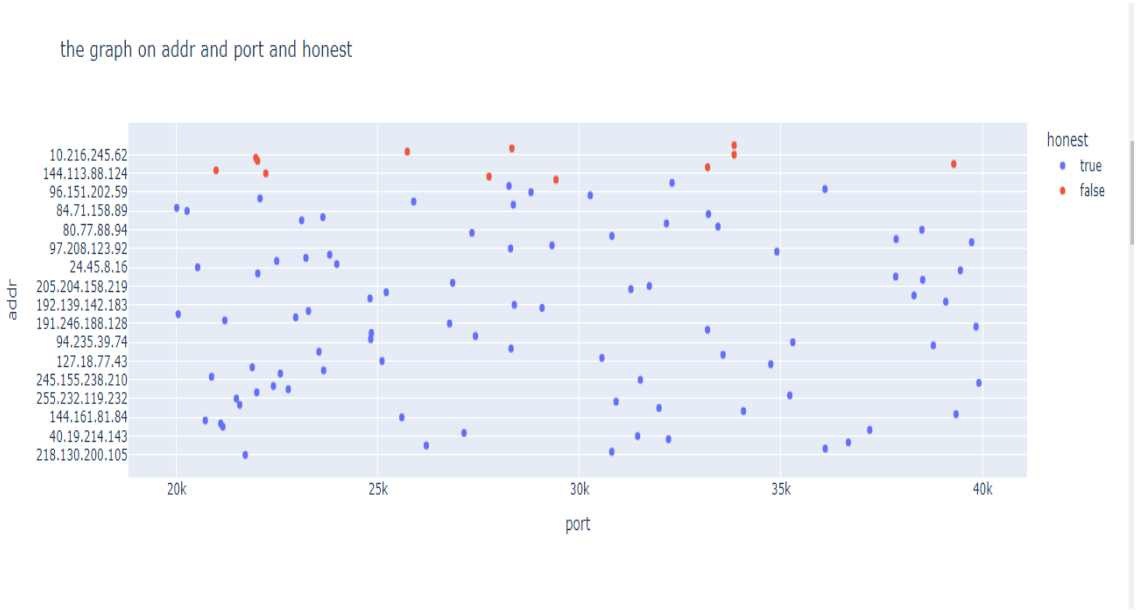


Figure 6.4: **Address, Port, Honest Data visualization**

In figure 6.4 provides the honesty factor in correspondence with address and port.

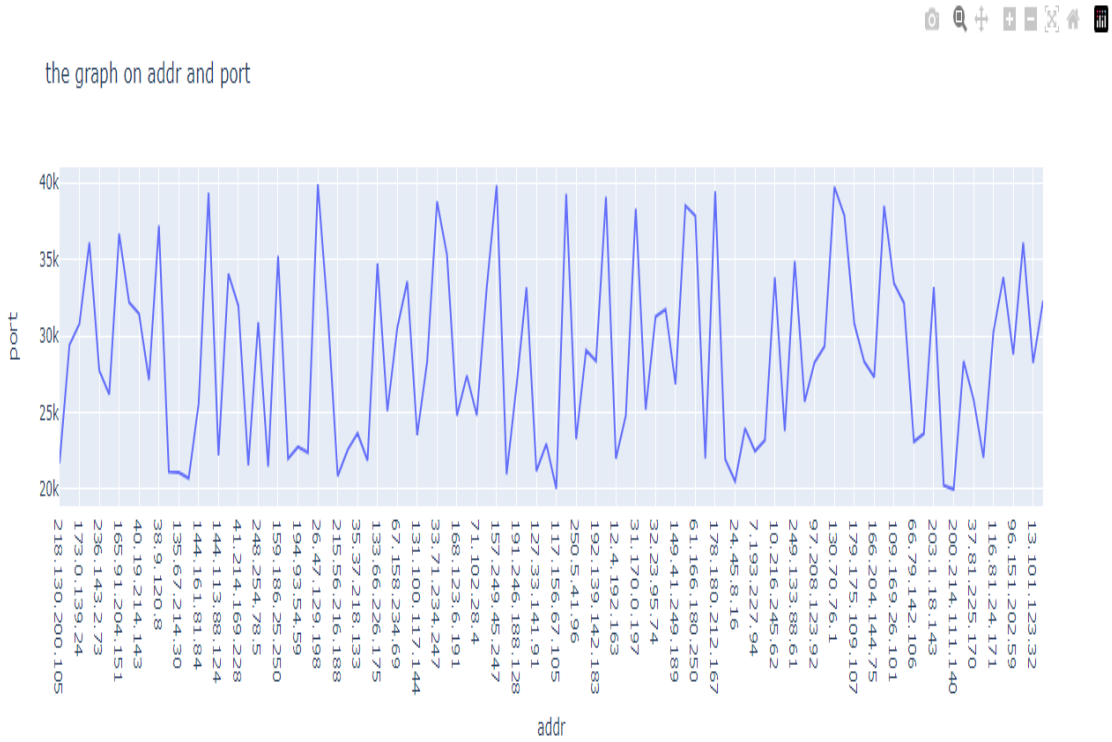


Figure 6.5: **Address Port Data visualization**

In figure 6.5 shows the interactive visualizations can also be used to simulate the performance of different consensus mechanisms under different conditions.A diagram for port data visualization in the proposed project can provide a visual representation of the data posted by IoT devices connected through satellite communication. The diagram can be designed to show different types of data, such as address , port and display them in real-time

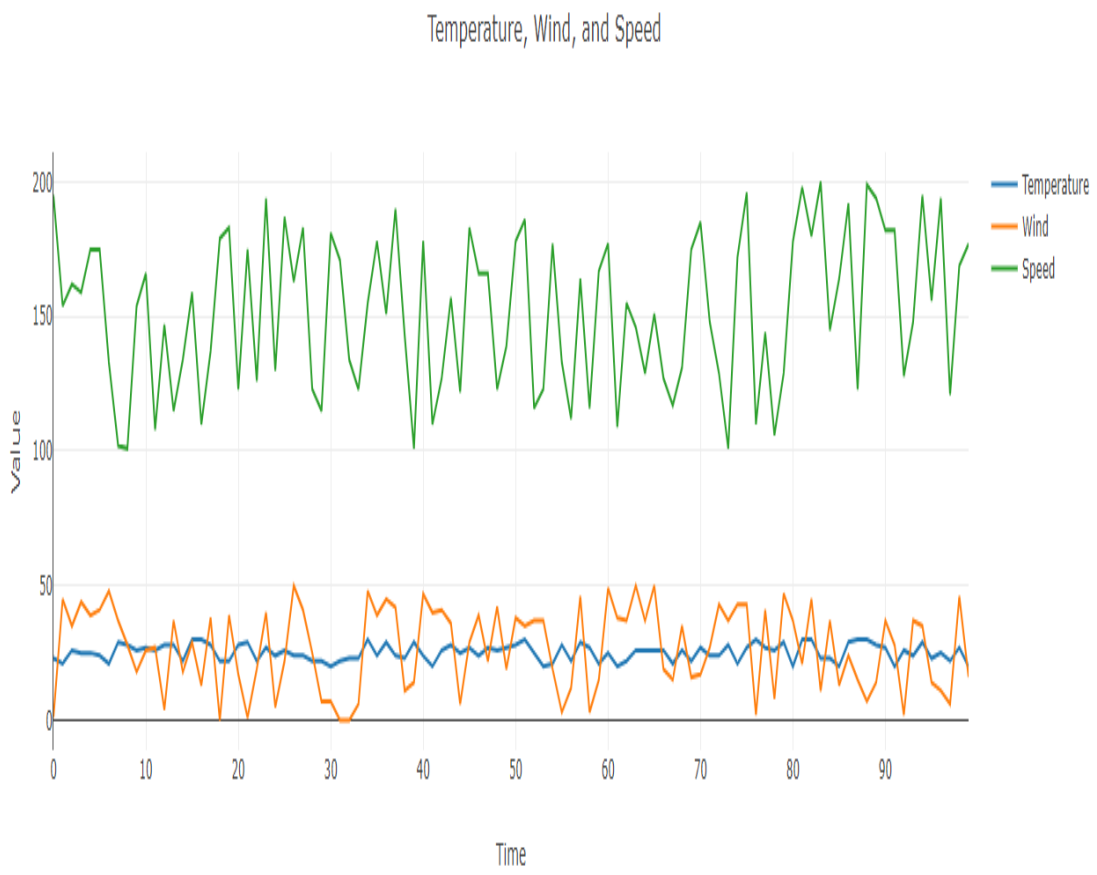


Figure 6.6: **Temperature, Wind, Speed Data visualization**

In figure 6.6 shows the display the time data on the x-axis and value data on Y-axis and the performance metrics of the consensus mechanism.The diagram can provide a comprehensive view of the relationship between temperature, wind, and speed data posted by IoT devices connected through satellite communication, enabling users to analyze the data and make informed decisions.

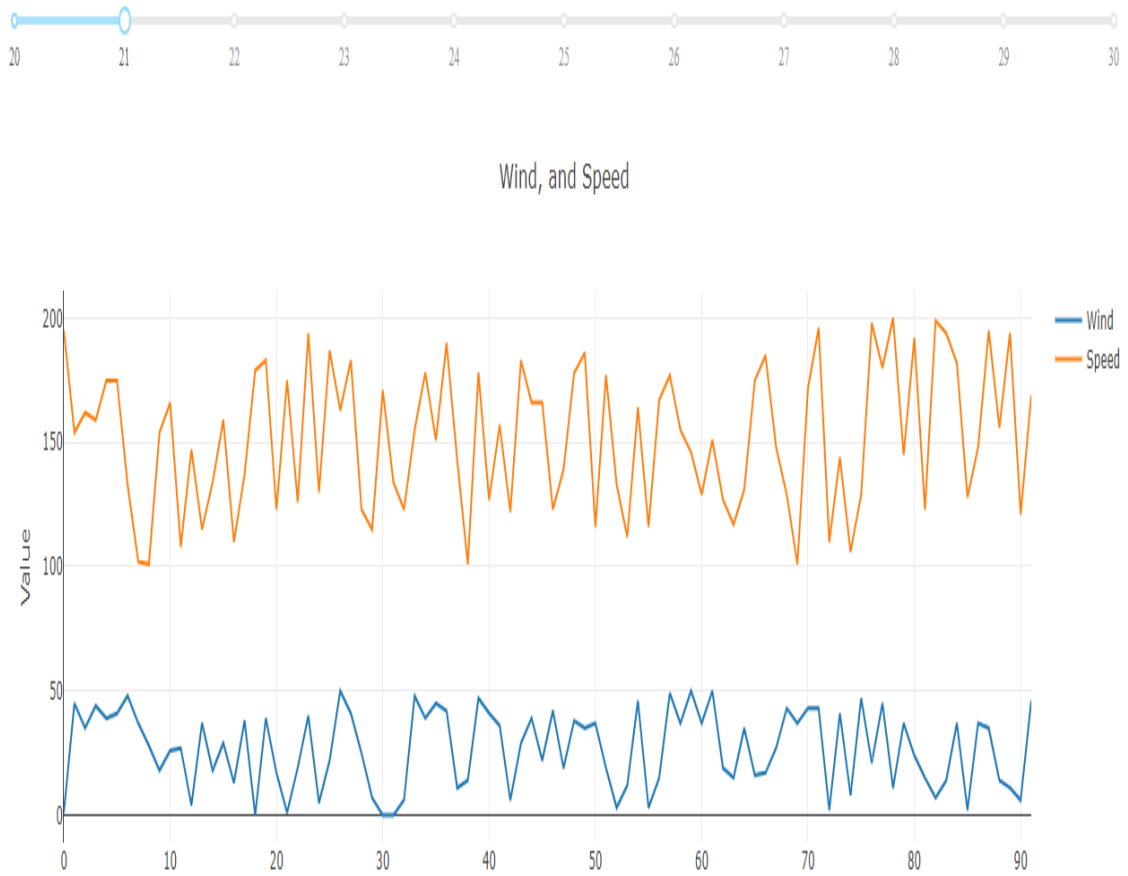


Figure 6.7: **Wind, Speed Data visualization with slider**

In figure 6.7 shows the line graphs for wind and their corresponding speeds is provided. Along with this a slider is attached for simulating the same for various values. By incorporating user interactions and alerts, the diagram can also help users quickly identify issues or anomalies in the data, improving the reliability and security of the system.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The IoT consensus mechanism for satellite communication project proposes a solution to the unique challenges of using satellite communication for IoT applications, such as low bandwidth, high latency, and the need for efficient and scalable consensus mechanisms. The comparative analysis of different consensus mechanisms presented in this project, including PoS, DPoS, FBA, and DAG, provides valuable insights into the selection of a suitable consensus mechanism for IoT satellite communication. The proposed system has the potential to improve the efficiency, scalability, and security of IoT applications in satellite networks. The project also provides guidelines for code standards and policies, as well as hardware and software specifications, to ensure code quality and efficient use of resources. Overall, the project contributes to the development of efficient and scalable IoT applications in satellite networks, which has the potential to impact a range of industries, including transportation, agriculture, and healthcare.

7.2 Future Enhancements

The IoT consensus mechanism for satellite communication project could include integrating with more satellite communication protocols to increase device support, implementing a hybrid consensus mechanism for improved performance, and optimizing the consensus algorithm for better energy efficiency, scalability, and security. Other potential enhancements could involve improving the user interface and adding more advanced data visualization tools to enhance the monitoring and analysis of communication performance.Additionally, exploring the use of machine learning algorithms could lead to more accurate consensus decisions.

Chapter 8

INDUSTRY DETAILS

8.1 Industry name

La Trobe University, Melbourne Victoria Australia

8.1.1 Duration of Internship (From date - To Date)

December, 2022 - May, 2023

8.1.2 Duration of Internship in months

4 months

8.1.3 Industry Address

La Trobe University,
Plenty Road and,
Kingsbury Dr,
Bundoora VIC 30864
Australia

8.2 Internship offer letter

Fwd: Meeting with - Vel Tech students & Zach Auhl - LTU - student research collaboration - IoT Consensus Mechanism Inbox ×

Student Mobility - Vel Tech, Chennai <studentmobility@veltech.edu.in> Thu, Aug 4, 2022, 3:46 PM

to Chennai, bcc: me

Aug 5 Fri

from: **Student Mobility - Vel Tech, Chennai** <studentmobility@veltech.edu.in>
to: "Student Mobility - Vel Tech, Chennai" <studentmobility@veltech.edu.in>
bcc: vtu15600@veltech.edu.in
date: Aug 4, 2022, 3:46 PM
subject: Fwd: Meeting with - Vel Tech students & Zach Auhl - LTU - student research collaboration - IoT Consensus Mechanism
mailed-by: veltech.edu.in
signed-by: veltech-edu-in.20210112.gappssmtp.com
security: Standard encryption (TLS) Learn more
Important according to Google magic.

Dear Students,

This is regarding

Date: 05.08.2022
Time: 11:30 AM to 12:30 PM

You can find the meeting link below. Please prepare well and present accordingly.

zoom

Hi there,

Michelle Selbie is inviting you to a scheduled Zoom meeting.

Join Zoom Meeting

Phone one-tap: Australia: +61370182005,,83323223884# or +61731853730,,83323223884#

Meeting URL: https://latrobe.zoom.us/j/83323223884
Meeting ID: 833 2322 3884

Join by Telephone

For higher quality, dial a number based on your current location.

Chapter 9

PLAGIARISM REPORT

Plagiarism Scan Report

Report Generated on: Apr 08,2023

Plagiarised

Unique

Total Words:	731
Total Characters:	5020
Plagiarized Sentences:	2.94
Unique Sentences:	39.06 (93%)

Content Checked for Plagiarism

Chapter 3
PROJECT DESCRIPTION
3.1 Existing System
The existing system for IoT consensus mechanisms involves using traditional consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS) for IoT devices connected through standard communication channels like the internet. However, these consensus mechanisms may not be suitable for IoT devices connected through satellite communication due to the unique challenges presented by satellite communication, such as high latency and limited bandwidth.Some existing systems for IoT satellite communication use centralized solutions where a single entity controls the network and makes decisions. However, centralized systems can present challenges such as security vulnerabilities, potential for single point of failure, and lack of transparency.
3.2 Proposed System
The proposed system for IoT consensus mechanism involves exploring and developing consensus mechanisms that are specifically designed for IoT devices connected through satellite communication. These mechanisms must be able to address the unique challenges presented by satellite communication, including high latency and limited bandwidth, while maintaining decentralization, security, and transparency.The proposed system will involve the use of IoT sensors connected through satellite communication to collect and transmit data to a central server. The central server will use a consensus mechanism to validate and authenticate the data received from the IoT sensors. The validated data will then be stored in a distributed ledger to ensure data integrity, transparency, and immutability.
4
3.3 Feasibility Study
A feasibility study is an important step in any project as it determines the project's viability and identifies potential risks and challenges that may arise during implementation.

Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```
from ipaddress import IPv4Address
import uuid
import random
import copy
import pandas as pd
from pprint import pp
import dash
from dash import dcc,html
from dash import Input, Output
from dash import dash_table
import plotly.express as px
import plotly.graph_objs as go
from flask import Flask, request,jsonify
from pymongo import MongoClient
import requests

client = MongoClient('mongodb://localhost:27017/')

# select the database and collection
db = client['major']
collection = db['data']

data = collection.find()
nodes=[]
messages=[]
for d in data:
    d.pop('_id')  # remove the '_id' field from each document
    if 'port' in d:
        nodes.append(d)
    else:
        messages.append(d)
df1 = pd.DataFrame(nodes)
df1['peers']=df1['peers'].apply(', '.join)
df = pd.DataFrame(messages)
df = pd.concat([df.drop(['payload'], axis=1), df['payload'].apply(pd.Series)], axis=1)
```

```
merged_df = pd.merge(df, df1, left_on='sender', right_on='id')
merged_df = merged_df.drop('id', axis=1)
app = dash.Dash(__name__)
fig = px.scatter(df1, x='port', y='addr', color='peers')
fig2 = px.scatter(df1, x='port', y='addr', color='honest',title='the graph on addr and port and
    honest')
fig3 = px.line(df1, x='addr', y='port',title='the graph on addr and port')
fig1 = px.scatter(df, x='sender', y='temperature',title='the graph on sender and temperature')

@app.server.route('/store', methods=['POST'])
def insert_many_data():
    data_list = request.get_json()
    result = collection.insert_one(data_list)
    return jsonify({'success': True})
@app.server.route('/get', methods=['GET'])
def get_data():
    # get data from the collection
    data = collection.find()

    # convert data to a list of dictionaries
    data_list = []
    for d in data:
        d.pop('_id')  # remove the '_id' field from each document
        data_list.append(d)

    # return the data as a JSON response
    return (data_list)

app.layout =dash.html.Div(children=[
    dash.html.H1(['The Dashboard'],style = {'textAlign':'center'}),
    dash.html.H2('Nodes Data'),
    dash.dash_table.DataTable(
        data=df1.to_dict('records'),
        columns=[{"name": i, "id": i} for i in df1.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    ),
    html.H3("Select the honest"),
    dcc.Dropdown(
        id='my-dropd',
        options=[{'label': temp, 'value': temp} for temp in df1['honest'].unique()],
```

```
        value=[df['temperature'].min()],
        multi=True
    ),
    html.Div(id='gr'),
    dash.html.H2('the graph between the port and peer'),
    dash.dcc.Graph(
        id='graph',
        figure=fig
    ),
    dcc.Graph(
        id='honest-graph',
        figure={
            'data': [
                {
                    'labels': ['Honest', 'Dishonest'],
                    'values': [df1['honest'].value_counts()['true'], df1['honest'].value_counts()['
                        false']],
                    'type': 'pie',
                    'name': 'honesty'
                }
            ],
            'layout': {
                'title': 'Honesty',
                'xaxis': {'title': 'Honest/Dishonest'},
                'yaxis': {'title': 'Count'}
            }
        }
    ),
    dash.dcc.Graph(
        id='graph1',
        figure=fig2
    ),
    dash.dcc.Graph(
        id='graph3',
        figure=fig3
    ),
    dash.html.H1('The messages'),
    dash.dash_table.DataTable(
        data=df.to_dict('records'),
        columns=[{"name": i, "id": i} for i in df.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
```

```
        page_size=8,
    ),
    dash.dcc.Graph(
        id='graphmess',
        figure=fig1
    ),
    dash.dcc.Graph(
        id='temperature-wind-speed-graph',
        figure={
            'data': [
                go.Scatter(name='Temperature', x=list(range(len(df['temperature']))), y=df['temperature'], mode='lines'),
                go.Scatter(name='Wind', x=list(range(len(df['wind']))), y=df['wind'], mode='lines'),
                go.Scatter(name='Speed', x=list(range(len(df['speed']))), y=df['speed'], mode='lines')
            ],
            'layout': go.Layout(
                title='Temperature, Wind, and Speed',
                xaxis=dict(title='Time'),
                yaxis=dict(title='Value')
            )
        }
    ),
    dash.dcc.Graph(
        id='wind-speed-graph',
        figure={
            'data': [
                go.Scatter(name='Wind', x=list(range(len(df['wind']))), y=df['wind'], mode='lines'),
                go.Scatter(name='Speed', x=list(range(len(df['speed']))), y=df['speed'], mode='lines')
            ],
            'layout': go.Layout(
                title=' Wind, and Speed',
                xaxis=dict(title='Time'),
                yaxis=dict(title='Value')
            )
        }
    ),
    dcc.Slider(
        id='temperature-slider',
        min=df['temperature'].min(),
        max=df['temperature'].max(),
        step=1,
        value=df['temperature'].min(),
        marks={str(temperature): str(temperature) for temperature in df['temperature'].unique()}
    ),
    dcc.Graph(id='graph'),
    dcc.Slider(
        id='temperature',
        min=df['temperature'].min(),
```

```
        max=df['temperature'].max(),
        step=1,
        value=df['temperature'].min(),
        marks={str(temperature): str(temperature) for temperature in df['temperature'].unique()}
    ),
    dcc.Graph(id='grap'),
    dcc.Dropdown(
        id='my-dropdown',
        options=[{'label': temp, 'value': temp} for temp in df['temperature'].unique()],
        value=[df['temperature'].min()],
        multi=True
    ),
    html.Div(id='gra'),
    html.H3("The combination of data"),
    dash.dash_table.DataTable(
        data=merged_df.to_dict('records'),
        columns=[{"name": i, "id": i} for i in merged_df.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    ),
    dash.dcc.Graph(
        id='honet-graph',
        figure=px.line(merged_df, x='sender', y='honest',title='the graph on Sender and honest')
    ),
])
@app.callback(Output('graph', 'figure'), [Input('temperature-slider', 'value')])
def update_graph(temperature):
    filtered_df = df[df['temperature'] >= temperature]
    figure={
            'data': [
                go.Scatter(name='Wind', x=list(range(len(filtered_df['wind']))), y=filtered_df['wind'], mode='lines'),
                go.Scatter(name='Speed', x=list(range(len(filtered_df['speed']))), y=filtered_df['speed'], mode='lines')
            ],
            'layout': go.Layout(
                title=' Wind, and Speed',
                xaxis=dict(title='Time'),
                yaxis=dict(title='Value')
            )
        }
```

```
    return figure
@app.callback(Output('grap', 'figure'), [Input('temperature', 'value')])
def update_graph(temperature):
    filtered_d = df[df['temperature'] >= temperature]
    figur={
            'data': [
                go.Scatter(name='Temperature', x=list(range(len(filtered_d['temperature']))), y=filtered_d['temperature'], mode='lines'),
                go.Scatter(name='Wind', x=list(range(len(filtered_d['wind']))), y=filtered_d['wind'], mode='lines'),
                go.Scatter(name='Speed', x=list(range(len(filtered_d['speed']))), y=filtered_d['speed'], mode='lines')
            ],
            'layout': go.Layout(
                title='Temperature, Wind, and Speed',
                xaxis=dict(title='Time'),
                yaxis=dict(title='Value')
            )
        }
    return figur
@app.callback(Output('gra', 'children'),
              [Input('my-dropdown', 'value')])
def update_graph(selected_temperatures):
    filte = df[df.temperature.isin(selected_temperatures)]
    return dash_table.DataTable(
        data=filte.to_dict('records'),
        columns=[{"name": i, "id": i} for i in filte.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    )
@app.callback(Output('gr', 'children'),
              [Input('my-dropd', 'value')])
def update_graph(selected_temperatures):
    filt = df1[df1.honest.isin(selected_temperatures)]
    return dash_table.DataTable(
        data=filt.to_dict('records'),
        columns=[{"name": i, "id": i} for i in filt.columns],
        filter_action='custom',
        editable=True,
        sort_action="native",
        sort_mode="multi",
```

```
        column_selectable="single",
        row_selectable="multi",
        row_deletable=True,
        selected_columns=[],
        selected_rows=[],
        page_current= 0,
        page_size=8,
    )
if __name__ == '__main__':
    app.run_server(debug=True)

#Api call.py
from ipaddress import IPv4Address
import uuid
import random
import copy
import pandas as pd
from pprint import pp
from pymongo import MongoClient
import requests
import json

client = MongoClient('mongodb://localhost:27017/')

# select the database and collection
db = client['major']
collection = db['data']

random_ips = []

for _ in range(100):
    random_ips.append(str(IPv4Address(random.getrandbits(32))))

connected_nodes = copy.deepcopy(random_ips)

nodes = []
node_ids = []

##############################
# Randomly Generate 50 nodes #
##############################

for _ in range(100):
    node = {}
    addr = random.choice(random_ips)
    random_ips.remove(addr)

    id = str(uuid.uuid4())
    node_ids.append(id)
    node["id"] = id
```

```
    node["port"] = random.randint(20000, 40000)
    node["addr"] = addr
    node_connections = random.sample(connected_nodes, k=5)

    while addr in node_connections:
        node_connections = random.sample(connected_nodes, k=5)

    node["peers"] = node_connections

    is_node_good = random.choices([True, False], weights=(0.9, 0.1))[0]
    node["honest"] = "true" if is_node_good else "false"

    nodes.append(node)

#pp(nodes)

messages = []

################################
# Randomly generate 50 messages #
################################

for _ in range(100):
    message = {}
    sender_receiver = random.sample(node_ids, k=2)
    message["sender"] = sender_receiver[0]
    message["receiver"] = sender_receiver[1]
    message["payload"] = {
        "temperature": random.randint(20, 30),
        "wind": random.randint(0, 50),
        "speed": random.randint(100, 200),
    }
    messages.append(message)

#pp(messages)

for item in messages:
    # convert the dictionary to a string and replace single quotes with double quotes
    json_string = str(item).replace("'", "\"")
    json_string = json.loads(json_string)
    response = requests.post('http://127.0.0.1:8050/store', json=json_string)
for ite in nodes:
    # convert the dictionary to a string and replace single quotes with double quotes
    json_strin = str(ite).replace("'", "\"")
    json_strin = json.loads(json_strin)
    response = requests.post('http://127.0.0.1:8050/store', json=json_strin)
```

10.2 Poster Presentation

IOT CONSENSUS MECHANISM

Department of Computer Science & Engineering
School of Computing
1156CS701 – MAJOR PROJECT
WINTER SEMESTER 2022-2023

ABSTRACT

The abstract for this project is to develop a dashboard that can visualize statistics and the network graph, gather and calculate statistics from IoT devices, and provide real-time visualization of the blockchain and consensus process. The project involves a decentralized network of Raspberry Pi running blockchain and consensus software developed by the project team. The monitoring server will collect and analyze data from the blockchain and consensus process, and the frontend system will display the data in an easy-to-understand format. The statistics collected from IoT devices will help ensure the network is functioning correctly and enable users to detect and address any issues. The overall goal is to create a comprehensive monitoring system that provides valuable insights into the health and status of the network and enables users to take appropriate action as needed.

Vtu15600/Abiram G
9347793901
vtu15600@veltech.edu.in

INTRODUCTION

The rise of blockchain technology has ushered in a new era of decentralization and transparency. However, monitoring the health and status of a blockchain network can be a challenge, particularly when it comes to large-scale deployments. To address this challenge, our project aims to implement a comprehensive monitoring system for a decentralized network of Raspberry Pi running blockchain software and consensus software.

The monitoring system will collect and analyze data from the blockchain and consensus process, as well as from IoT devices connected to the network. The data will be displayed in real-time using a visually appealing frontend system, providing users with valuable insights into the health and status of the network. Our project represents a significant step forward in the world of blockchain monitoring and management, providing a comprehensive and user-friendly solution for keeping tabs on blockchain networks.

METHODOLOGIES

The first scatter plot shows the relationship between the port number and the IP address of each node, with the color of each node indicating its connections to other nodes. The second scatter plot shows the relationship between the sender and receiver of each message, with the color indicating the temperature of the message payload.

The two tables show the data for the nodes and messages, respectively. The table for nodes includes columns for the node ID, port number, IP address, connected peers, and whether the node is honest or not. The table for messages includes columns for the sender, receiver, and various attributes of the message payload, such as temperature, wind, and speed.

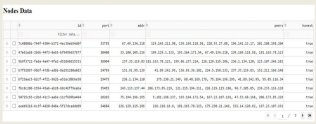
RESULTS

A comprehensive monitoring system for a decentralized network of Raspberry Pi devices running blockchain and consensus software. The system would gather and calculate statistics from IoT devices and provide real-time visualization of the blockchain and consensus process. The monitoring server would collect and analyze data from the network and the frontend system would display the data in an easy-to-understand format.

The system would help ensure the network is functioning correctly, detect and address any issues, and provide valuable insights into the health and status of the network. It would enable users to take appropriate action as needed to maintain the network's performance and security.

Output 1.The Graph between port and peers.

Output 2 Node Data:

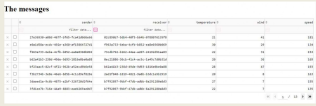


STANDARDS AND POLICIES

1.Software Development Standards: The project team should follow best practices for software development, including code reviews, version control, and automated testing. They should also document their code and follow coding standards to ensure the software is maintainable and scalable.

2.Accessibility Standards: The frontend system should adhere to accessibility standards such as WCAG 2.0 to ensure that it is usable by people with disabilities

3.Network Monitoring Policy: The project team should define a policy for network monitoring that outlines the types of data that will be collected, how it will be analyzed, and who will have access to it



CONCLUSIONS

This project aims to develop a custom blockchain and consensus mechanism using Raspberry Pi connected in a decentralized network. To monitor the status of the blockchain and consensus process, a centralized monitoring server will be built using Grafana as the frontend, Prometheus as the backend, and a custom Python REST API for each Raspberry Pi to provide data. The implementation of this technology will enable the visualization of statistics, network graphs, and the consensus process in real time.

ACKNOWLEDGEMENT

1. Dr. V .Dhilip kumar M.E .,Ph.D ,
2. +91 9994614463
3. vdhilipkumar@veltech.edu.in

References

[1] Arul, P. and Renuka, S. (2021). Blockchain technology using consensus mechanism for IoT-based e-healthcare system. IOP Conf. Ser.: Mater. Sci. Eng., 1055(1), 012106.

[2] Auhl, Z.; Chilamkurti, N.; Alhadad, R.; Heyne,W. A (2022). Comparative Study of Consensus Mechanisms in Blockchain for IoT Networks. Electronics MDPI, 11, 2694.

[3] Cao, B., Li, Y., Zhang, L., Zhang, L., Mumtaz, S., Zhou, Z., Peng, M. (2019). When Internet of Things Meets Blockchain: Challenges in Distributed Consensus. IEEE Network, 33(5), 234-240.

[4] Huang, J., Kong, L., Chen, G., Cheng, L., Wu, K., Liu, X. (2019). B-IoT: Blockchain Driven Internet of Things with Credit-Based Consensus Mechanism. IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 1348.

[5] Li, C., Zhang, J., Yang, X., Youlong, L. (2021). Lightweight blockchain consensus mechanism and storage optimization for resource-constrained IoT devices. Information Processing and Management, 58, 102602.

[6] Raghav, Nitish Andola, S. Venkatesan, Shekhar Verma (2020). PoEWAL: A lightweight consensus mechanism for blockchain in IoT, Pervasive and Mobile.

[7] Salimitari, M. and Chatterjee, M. (2019). A Survey on Consensus Protocols in Blockchain for IoT Networks, arXiv:1809.05613v2 [cs.NI], pp. 1-21.

[8] Tsang, Y.P., Choy, K.L., Wu, C.H., Ho, G.T.S. Lam, H.Y. (2019). Blockchain-driven IoT for Food Traceability with an Integrated Consensus Mechanism. IEEE Access, Vol. XX, pp. 1-1.

[9] Wang, C., Tan, X., Yao, C., Gu, F., Shi, F., Cao, H. (2022). Trusted Blockchain-Driven IoT Security Consensus Mechanism. Sustainability, 14(9), 5200.

[10] Wu, Y., Song, L., Liu, L., Li, J., Li, X., Zhou, L. (2020). Consensus Mechanism of IoT Based on Blockchain Technology. Sensors, 20(19), 5636.

[11] Yuan, X., Luo, F., Haider, M. Z., Chen, Z., Li, Y. (2021). Efficient Byzantine Consensus Mechanism Based on Reputation in IoT Blockchain. Open Access Library Journal, 8(5), 1-12.

[12] Yue Wu, Liangtu Song, Lei Liu, and Linli Zhou (2021). Blockchain-based IoT Consensus Mechanism System, IOP Conf. Ser.: Earth Environ. Sci. 769, 042033.