KLE Society's
KLE Technological University

A Capstone Project Report

On

A SECURE AND EFFICIENT CONSENSUS ALGORITHM FOR BLOCKCHAIN NETWORKS

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering in

Computer Science and Engineering

submitted by:

JYOTHI S HOSAMANI	**01FE16BCS083**
SHRIYA HIREHOLI	**01FE16BCS079**
ARPITA V KUSABI	**01FE16BCS240**

Under the guidance of

DR. NARAYAN D G

.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI – 580031 (India).

KLE Technological University

Creating Value
Leveraging Knowledge

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Capstone Project entitled "A Secure and E□cient Consensus Algorithm for Blockchain Networks" is a bonafied work carried out by the student team Ms.ShriyaHireholi–01FE16BCS079, Ms. Jyothi S Hosamani–01FE16BCS083, Ms.Arpita V Kusabi–01FE16BCS240, in partial fulfillment of completion of 8th semester B. E. in Computer science and Engineering during the academic year 2019 – 2020. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Name of the Guide

Dr. Narayan D G

Head of SoCSE

HEAD
SCHOOL OF COMPUTER SCI...
& ENGINEERING
KLE Technological University
HUBBALLI-580 031

Name of the examiners

1.

2.

Signature with date

1.

2.

ABSTRACT

Blockchain era seems to have a broad scope of programs under financial services, Industrial products, Healthcare, Media etc. Consensus mechanism being the center of blockchain, affects the safety and efficiency of blockchain without delay. More than 30 consensus algorithms such as Proof-of-Work (PoW), Proof-of-Stake (Pos), Ripple Protocol Consensus Algorithm (RPCA), Proof- of-Authority (PoA), Byzantine Fault Tolerance(BFT) etc have existed in recent years. But these algorithms lag far behind in security, stability and operating efficiency. Most of the well-known crypto-currencies use Proof of Work consensus algorithm. However, it lacks sustainability and is not suitable for blockchain-based security solutions. Proof of Stake (PoS) is more vulnerable to who has enough stakes to destroy the system by investing stake and there is "the rich get richer" problem. To resolve these problems we propose an algorithm called Delegated Proof-of-Stake (DPoS) which breaks the cycle of competition in computing resources in generating blocks. In this algorithm we introduce an election system based totally on PoS and reduce the cost of producing a block. This paper also introduces an advanced DPOS consensus algorithm called as Delegated Proof-of-Stake with Downgrade (DDPoS). A downgrade system is introduced to remove the corrupt nodes to maintain security. The performance analysis shows that the proposed consensus algorithm is substantially more effective than other consensus algorithms.

Keywords : *Blockchain, Consensus Mechanism, Proof-of-Work, Proof-of-Stake, Delegated Proof-of-Stake, Delegated Proof-of-Stake with Downgrade, Downgrade mechanism, Efficiency.*

ACKNOWLEDGEMENT

We have been bestowed the privilege of expressing our gratitude to everyone who helped us in completing the dissertation work.

We take this opportunity to thank Dr. Ashok Shettar, Vice-chancellor, KLE Tech and to Dr. Prakash Tewari, Principal, B V Bhoomaraddi College of Engineering And Technology, Hubli.

We also take this opportunity to thank Dr.Meena S M, Professor and Head of Department, Department of Computer Science and Engineering for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

We sincerely thank our guide Dr. Narayan D G , Department of Computer Science and Engineering for his guidance, inspiration and wholehearted co-operation during the course of completion.

Also the research papers mentioned in the literature survey helped us in understanding the present implemented consensus algorithms used in building a Blockchain and their advantages and disadvantages which has been further used to build an efficient consensus algorithm.

We would also like to thank our esteemed KLE Technological University for providing us such an opportunity for gaining knowledge and learning new things. We would like to thank our supporting staff for providing us basic needs to complete our project and finally our thanks and appreciations go to our colleague in providing us insight and ideas for project.

Finally, we thank one and all who have directly and indirectly assisted us in the project work.

Shriya Hireholi - 01FE16BCS079
Jyothi S Hosamani - 01FE16BCS083
Arpita V Kusabi - 01FE16BCS240

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

A blockchain is a distributed ledger that statistics transactions throughtout a network of nodes. Blockchain is a type of data structure, composed of data blocks in the form of a linked list, using a distributed ledger with cryptographic techniques to assure authenticity and security of transactions information. With the continuous innovatio of blockchain technology, the blockchain can plays a good role in Artificial Intelligence, Internet of Things, Education Reform, Medical systems and other fields.

Consensus mechanism has been a maximum important issue of the whole blockchain network, because its efficiency directly determines the performance of the blockchain. There will be many challenges and issues in the process of implementing blockchain technology, including how to design an effective consensus protocol. With the continuous improvement of blockchain network, the consensus mechanism is rapidly being adapted from the old Proof-of-Work (PoW) to the new Proof-of-Stake (PoS), Practical Byzantine Fault Tolerance (PBFT) and some different consensus algorithms such as Proof of Activity (PoA), Proof of Burn (PoB), Proof of Luck (PoL), and Stellar Consensus Protocol (SCP) as per the emerging requirements. However, all these algorithms have some major shortcomings.

In Proof of Work (PoW), every node goes head to head for the desire to produce blocks with the aid of fixing computing puzzle and via possessing more computational assets than any others, that now not only substantially uses the computational power assets but additionally leads to the inefficiency. PoW's key characteristic is its asymmetry: the work must be difficult for the recipient, but easy for the service provider to verify. PoS aims to address the problem of energy consumption created by PoW. To this end, Proof of Stake replaces PoW's engagement by choosing stakeholders by the amount of tokens they stake, to append to the blockchain. The proprietor of the chosen token then receives the chance to append to the department and simultaneously collect a reward. Byzantine Fault Tolerance(BFT) is designed for stable device replication that tolerates Byzantine faults. Each client in the network eventually receives replies t their requests and those replies are correct in keeping with linearizability. Proof of Activity (PoA) is a hybrid between PoS and PoW. Blocks are generated through PoW mining mechanisms and then a PoS style mechanism occurs where validator nodes stake tokens in order to be chosen to mine blocks DPoS proposes an election system based totally on PoS, and decrease the cost value of producing a block to 3 sec. It splits the peers in the blockchain network into two groups: delegates and workers. The main function of the worker node is to

create the transaction while the delegate node is to produce blocks in round robin fashion and verify the blocks.

Compared to other consensus algorithms DPOS has verified robust, secure, and efficient through years of reliable operation on more than one blockchains. In any blockchian malicious nodes might exist. The malicious nodes are the ones that fraudulently break the reliable consensus protocol, manipulate transaction details, cause network traffic and interrupt the network's normal performance. The blockchain system eventually turns unsafe, unstable and inadequate. To solve these issues, we introduce an enhanced consensus algorithm DDPOS, which introduces the idea of PoW to enhance equity, and the ideology of DPoS to lower utilization of assets and enhance the blockchain system's consensus efficiency. The key innovation of the enhanced algorithm is that we propose a downgrade mechanism for rapid downgrading of malware nodes and upgrading accurate nodes to maintain system security and good functioning.

1.1 Literature Review / Survey

In this section, we will know what is a literature survey and the significance it poses before undertaking any project. It also presents with the literatures of various authors surveyed based on the techniques used, their advantages, programming platforms used, issues faced, etc. that helped during the implementation of the given project.

One form of review article is a literary or narrative analysis. A literature survey is a research paper containing the latest information, including empirical findings, as well as conceptual and analytical accomplishments to a given subject. Literature survey is a secondary research method and no new or original experimental work is recorded. These are most commonly linked to scholarly literature, these surveys can be based on older studies published, and not to be baffled with blog posts that might also appear in the same journal. Literature surveys for research basis in nearly all the academic discipline. A small-scale literature survey can be published in a peer-reviewed journal article detailing new work, suitable for positioning the current study within the relevant literature and providing context to the reader. In this kind of case, the analysis generally precedes work methodology and results sections. Producing a literature survey is a part of under-graduate and post-graduate student work, including preparing a proposal, a dissertation or an article in a research paper. Literature surveys are just obvious in paper proposals or prospectuses.

The main types of literature surveys are: evaluative surveys, exploratory surveys, instrumental and the systematic surveys, which is often classified differently. The systematic surveys focuses on research, attempting to identify, assess, select and synthesize all the research proof relevant to this question. A meta analysis is usually a systematic survey which uses statisti-

cal methods to effectively aggregate the data used in all selected studies to produce a more effective result.

1.1.1 Introduction to Blockchain

In literature [1], the author explained about blockchain. Blockchains are tamper proof and tamper resistant digital ledgers implemented in a distributed fashion and usually without a central authority. At their basic level, they permit a network of customers to record transactions in a shared ledger inside that community, such that underneath regular operation of the blockchain network no transaction can be changed as soon as published. in 2008, the blockchain concept became combined with numerous other technologies and computing standards to create current crypto-currencies electronic coins protected thru cryptographic mechanisms instead of a relevant repository or authority.

This generation became widely recognized in 2009 with the release of the Bitcoin network, the first of many modern-day cryptocurrencies. In Bitcoin and similar systems, the switch of digital records that represents electronic cash takes region in a distributed system. Bitcoin customers can digitally statistics this transfer publicly, allowing all members of the network to independently confirm the validity of the transactions. The Bitcoin blockchain is independently maintained and managed with the aid of a distributed institution of participants. This, combined with cryptographic mechanisms, makes the blockchain immune to subsequent attempts to change the ledger(modify blocks or duplicate transactions). Blockchain technology has enabled the development of many cryptocurrency structures together with Bitcoin and Ethereum. Because of this, blockchain technology is often considered as bound to Bitcoin or in all likelihood cryptocurrency solutions in general. The technology, indeed, is applicable for a wide range of applications, and is being studied for a number of sectors.

The numerous additives of blockchain generation along with its reliance on cryptographic primitives and distributed systems can make it tough to understand. However, every component can be decsribed sincerely and used as a building block to apprehend the larger complex system. The blockchains can be described informally as:

Blockchain is a shared digital ledgers of transactions that are approved cryptographically, organized into blocks. After validation and undergoing a consensus decision each block is cryptographically linked to the previous one(making it apparent). New blocks are added, older blocks get harder to modify(creating resistance to tampering). New blocks are replicated throughout copies of the ledger within the network,and any conflicts are automatically resolved using established rules. Figure 1.1 shows the basic blockchain structure.

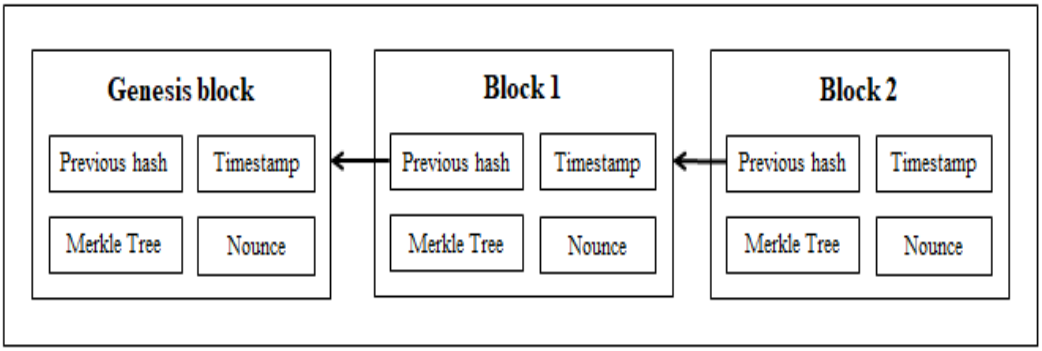

Figure 1.1: Blockchain Structure

Blockchain Types

In literature [2], the author explains about different types of blockchain and classifies into three categories.

A Based on the nature of data accessibility Blockchain can be categories as below:

i Public Blockchain: In this type of blockchain, anybody can read and submit transactions.

ii Private Blockchain: In this type of blockchain only one organization or all subsidiary organization within the same group are permitted to read and submit transactions.

iii Community/Consortium Blockchain: In this type of blockchain, multiple group of organizations form a consortium and are allowed to submit transactions and read transactional data.

iv Hybrid Blockchain: This is new type where any of the three Public, Private or Community/ Consortium, blockchain can be combined to ease transactions. This type is used to configure a blockchain platform in multimode.

Comparison between these blockchain types is shown in table 1.1.

B Based on the need of authorization to participate in Blockchain it can be categorized as below:

i Permissionless Blockchain: No prior permission is needed to perform in this type of Blockchain. Everyone is allowed to take part in verification process and can join blockchain network with their computational power.

Table 1.1: Comparison of Blockchain based on data accessibility

Properties	**Public Blockchain**	**Private Blockchain**	**Consortium Blockchain**
Consensus determination	All miners	One organization	Only selected set of nodes
Read Permission	Public	Public or restricted	Public or restricted
Write Permission	Anyone	Network operator only	Authorised participants
Immutability	Nearly impossible to modify	Can be modified	Can be modified
Efficiency	Low	High	High
Centralized	No	Yes	Partial
Consensus Process	Permissionless	Permissioned	Permissioned
Examples	Bitcoin, Litecoin, Ethereum	Company internal	R3, EWF, B3I

ii Permissioned Blockchain: To join this type of blockchain prior permission is necessary. Only authorized users are allowed to run nodes to verify transactions in blockchain network.

iii Hybrid Blockchain: There could be a possibility that a node is participating in Permissionless and Permissioned Blockchain together to encourage inter Blockchain communication such a blockchain can be called Hybrid Blockchain. A Blockchain platform can likewise be configured to support Permissioned and/or Permissionless model.

Comparison between these blockchain types is shown in table 1.2 and table 1.3.

C As far as core functionality and smart contract support in concern, blockchain can be categorized as below:

i Stateless Blockchain: This system only focuses on transaction advancement and chain functionality that is verifying the transaction by computing hashes. It is autonomous from smart contract logic layer thus unaffected from smart contract code bugs and vulnerabilities.

ii Stateful Blockchain: This system provides smart contract and transaction computing capabilities. It also supports multiaspect business logic, its optimization and maintains logic states.

Comparison between these blockchain types is shown in Table 1.4.

Table 1.2: Comparison between Permissioned and Permissionless Blockchain

Permissioned Blockchain	**Permissionless Blockchain**
Trusted	Trust-free
Faster	Slower
Central servers	Distributed servers
Token is needed	Token is not needed
Authorized access	Open access
Pre-determined trusted validators	Anonymous, fully decentralized validators
Hyperledger	Bitcoin, Ethereum

Table 1.3: Comparison between Permissionless, Permissioned, Public and Private Blockchains

Permissionless Public	**Permissioned Public**	**Permissionless Private**	**Permissioned Private**
PoW Bitcoin, Ethereum, Zcash etc	PoS Ethereum after Caspar	Federated Byzantine Agreement (FBA)	Multi-Signature
Anybody can download the protocol and verify transactions	Anybody who meets certain pre-specified criteria can download the protocol and verify transactions	Anyone can participate in the verification of transactions	Only its member can verify transactions

Table 1.4: Consensus algorithms used by various types of blockchains

Type of Blockchain	Consensus Algorithms	Nature
Public Blockchain	PoW, PoS, DPoS	Open and decentralized
Private Blockchain	PBFT, RAFT	Closed
Permissioned Blockchain	PBFT	Private or Public and Permissioned

Blockchain Protocol Stack

The blockchain protocol stack is divided into six layers: DataLayer, NetworkLayer, ConsensusLayer, IncentiveLayer, ContractLayer, and ApplicationLayer. The blockchain protocol stack is shown in Figure 1.2.

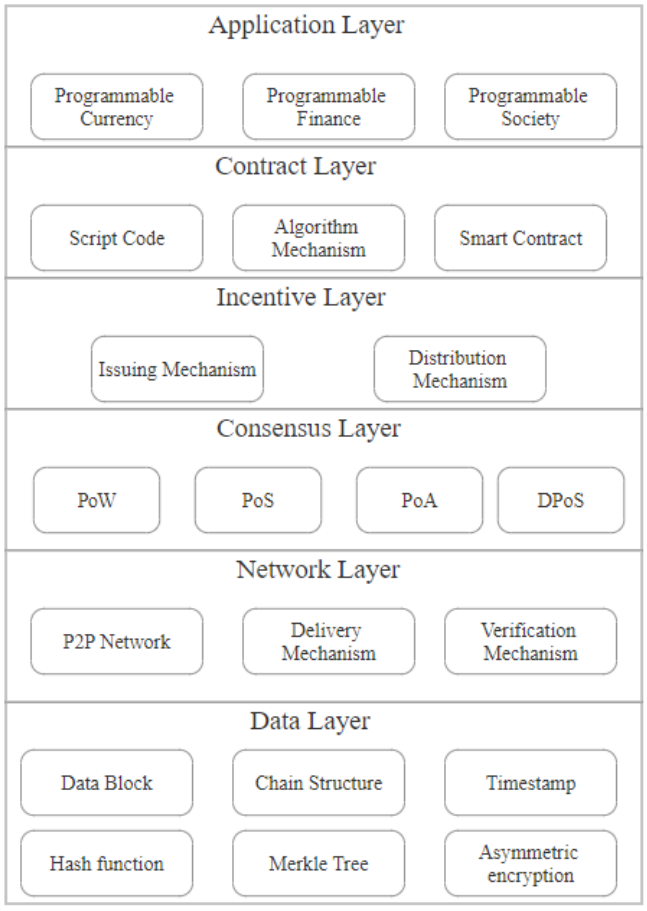

Figure 1.2: Blockchain Protocol Stack

Data Layer:- This layer presents the anchored information obstructs, alongside the related boundaries including symmetric encryption, timestamp, hash capacities and Merkle tree. Each processing hub that successes the agreement resistance will be constrained to fabricate another square in a blockchain organize. An information square is chiefly made out of a square header and a square body. The square header includes an adaptation number, going before square hash esteem, a Merkle tree of exchanges, a timestamp, a current difficulty level, and an irregular number. Exchange records are put away consistently in the square body. The squares are chains individually in consecutive request, shaping the spine from the first square to the more up to date one that holds the whole past information.

Network Layer:- This layer defines the dispensed networking, data forwarding, and verification mechanisms. All Peers are equally skilled, educated participants in the network without a central overseer or strict hierarchy. When a bit of evidence or a fresh block is produced, it is transmitted to the network [3]. Each blockchain node verifies the data received including fresh blocks according to pre-determined functions, rejects inaccurate ones and transmits others to adjacent peers within the network. In this manner, the data or blocks that pass validation from a large proportion of peers are added to the blockchain. Blockchain information is put away on every single hub and can be simulated and reestablished without complexities in all but one node, even in the most pessimistic scenario of disappointment [3].

Consensus Layer:- It contains all possible and available consensus algorithms. Different sorts of consensus protocols have been innovated in recent years. For example, the consensus-based proof-of-work mechanism is the most generally utilized protocol that asks peers to frequently execute cryptographic capacities or other customer quests to approve the information [4]. Proof-of-Stake (PoS) based consensus necessitates that the node with bigger measure of stake (ex: coin, token, etc) to approve the information [5]. Different algorithms incorporate Delegated Proof-of-Stake (DPoS), Proof-of-Authority and so on.

Incentive Layer:- The Incentive coordinates financial compensation into blockchains and defines the processes by which it is released and distributed to the winning node. If a fresh block is made, a specific measure of coins will be created as a prize for encouraging the network to continue its data verification efforts. Those coins will be distributed based on their investments to all nodes.

Contract Layer:- The contract layer contains different contents, protocols and smart contracts, which fill in as significant activators to the static information, cash, or resources put away in the blockchain. Smart contracts are a gathering of state-reaction decides that are put away on a blockchain and secured by the blockchain to self-verify, self-execute, and self-enforce. Typically speaking, if at least two gatherings agree to the entirety of the contract, they sign the smart contract cryptographically and communicate it to the system for validation. When the pre - specified terms are activated, smart contracts will auto-execute the agreement's precondition and conjure a comparing reaction activity without third party intervention.

Application Layer:- This layer contains potential application situations. The application layer is the business logic for various digital asset transactions.

1.1.2 Consensus Algorithms

The consensus mechanism is a protocol which provides how equilibrium is reached among all network consensus nodes, and establishes record validity. Blockchain network participants use this to reach an agreement whether the transaction is valid and synchronize it with the blockchain. Each consensus unit in the blockchain checks the transaction and affirms the data by algorithm. The data can be written into the blockchain after validation by a certain node density. Currently, the most famous consensus algorithms are: POW, POS, PoA and BFT. Since the advent of Bitcoin to now, there are much more than 30 consensus algorithms, most of which are based on the four consensus algorithms above.

Most of the blockchain technologies use Proof-of-Work (POW) consensus algorithm. The two most famous cryptocurrencies, Bitcoin and Ethereum, are POW based . Transactions in the blockchain needs to be verified which is done by miners. Miners compete each other to solve computational puzzle in order to mine a block. These miners need to be aware of the cryptographic hash value of the previously recorded block, which is forbidden from everyone. In order to find that the miner must follow the brute force approach which is a competitive process and needs immense computing power. The puzzle is updated every 14-15 days to make it more complicated and it is too difficult to carry out a DDoS attack by collecting 51 percent of the total processing power in this network. Hence this mechanism makes blockchain very secure but consumes more power i.e, strong safety comes at a costly price. The increasing computational capacity of the nodes needs more and more electrical resources, and the presence of all nodes in the process of transaction validation affects the scalability and transaction throughput, needs continuous upgradation of hardware to solve ever more complex mathematical puzzles. [4].

Fahad Saleh in 2017 [5] proposes Proof-of-Stake. PoS seeks to solve the problem of energy expenses created by PoW. To this end, Proof of Stake replaces PoW's engagement by choosing stakeholders by the amount of tokens they stake, to append to the blockchain. The simplest implementation of PoS involves selecting each blockchain branch from the universe of blockchain tokens uniformly and at random. The selection stake-holders can be done in two ways: either by selecting randomly or based on the amount of tokens each node is willing to stake, the one with more tokens staked is selected as stake-holder. The recipient of the specified token is then given the opportunity to connect to the branch and earns a reward at the same time. This protocol succeeds in limiting energy costs to nominal levels, but in doing so, threatens to re-set up the trouble solved by means of PoW, the Nothing-at-Stake problem. The advantage of PoS is that there is no forging process. This eliminates the need to solve

a strong cryptographic puzzle, and also the continuous enhance of the cost of manufacturing and soaring energy. The whole network ought not to be engaged in the process of transaction validation, which enhances the scalability. Disadvantages of PoS are: : Many of the coins or tokens can be purchased in the network, become the stake-holder of preference and verify the erroneous transactions as part of an offense, any stakeholder who is staking may very probably have malicious intentions. They are allowed to vote on both blockchain network versions.

In literature [6], Dantheman provides method where he divides the DPoS algorithm into two parts: election of block creaters and organize creation. The selection process ensures stake-holders are ultimately responsible because when the channel does not run smoothly stakeholders lose the most. Another method for Delegated Proof of Stake is by partitioning the peers in the blockchain network into three classifications: witness nodes, delegate nodes and worker nodes. Witnesses are the backbone of the entire system and are appointed by resource voting by all the nodes. The peers who win the greatest number of seats are witnesses and take turns to create blocks in a round robin fashion. The delegates can file an application to update blockchain even though they won't get paid. Workers are entitled to put in fresh projects and to earn credit from the projects elected.

The DDPoS algorithm consensus process is composed of three phases: electing consensus nodes, reaching consensus on block verification, and downgrading malicious nodes. In the election phase of consensus nodes, all nodes in the blockchain system are categorised to assign different tasks to different types of nodes, and are mainly divided into two types: consensus node (composed of witness nodes and candidate nodes) and trading node. The main aim of the consensus nodes is to produce and verify the blocks while that of trading nodes is to produce the transaction. The enhancements of PoW and DPoS can be merged to strengthen the original DPoS algorithm. Here PoW is used to select a set of nodes with sufficient computing power, rather than participating in the election. The algorithm then mandates that each node has only one vote to vote at random, which increases equity and eventually reduces the rights of block generation to prevent collusion attacks and increase node operation in the blockchain network. Finally, it implements a downgrade mechanism to rapidly downgrade the malicious nodes in order to maintain good system operation and safety.

Proof of Authority (PoA) is a consensus system based on reputation, which provides a realistic and efficient solution for blockchain networks. Some of advantages are: valid and trustworthy identities, difficulty to become a validator, standard for validator approval. PoA has high Byzantine Fault Tolerance compared to typical centralized systems without a distributed consensus mechanism. Disadvantages are: validators are available to all, knowing the identities of the validators may potentially lead to corruption by third parties, sacrificing decentralization in order to achieve high throughput and scalability.

In literature [7], the author explains about Practical Byzantine Fault Tolerance algorithm. BFT can be used to develop highly accessible systems tolerant of Byzantine faults. It shows

how to build Byzantine fault tolerant technologies that can be used in practical application of real services. They do not rely on unrealistic assumptions and they perform well. BFT works in asynchronous environments and incorporates mechanisms to defend against Byzantine-faulty clients, and recovers replicas proactively. The recovery mechanism in BFT allows the algorithm to tolerate any number of faults over the lifetime of the system. But less than one-third of the replicas become faulty within a small window of vulnerability. The window may increase under a denial-of-service attack but the algorithm can detect and respond to such attacks and can also detect when an attacker corrupts the state of a replica. BFT is implemented as a generic program library with a simple interface. BFT library can be used to build practical systems that tolerate Byzantine faults.

Comparision of all these consensus algorithms is shown in the table 1.5.

Table 1.5: Comparision of the famous blockchain consensus algorithms

Consensus Algorithm	**POW**	**POS**	**DPoS**	**PoA**	**PBFT**
Blockchain type	Permissionless	Both	Both	Permissioned	Both
Mechanism	Race to solve math problem	Percentage of token nodes stake	Stake-holders vote for delegates, who are chosen by what percentage of token they hold	Reputation and Identity	Repeat incoming message blocks to every node in the network
Rate of transaction	Low	High	High	Medium	High
Tolerance rate	$<$ 25% of computational power	$<$ 51% of the stake	$<$ 51% of validators	$<$ 25% of computational power	$<$33.3% of faulty replicas
Energy requirement	High	Medium	Low	High	Low
Node Identity	Open	Open	Open	Open	Permissioned
Network scalability	High	High	Medium	High	Medium
Examples	Bitcoin, Litecoin, Ethereum (until 2018)	Tendermint, Ethereum(from 2018)	Bitshares, EOS, Lisk, Steem, Ark	Decred, SIKKA	Practical BFT, Federated BFT

1.2 Overview of the project

All blockchains are essentially a deterministic state machine which transactions function in. Consensus is the method of finding consensus on a deterministic transaction order and eliminating invalid transactions. There are several different consensus algorithms which could generate equivalent transaction ordering. However, DPOS proved to be robust, safe and effective across years of reliable operation on multiple blockchains.

In addition to this, malicious users or attackers may exist in any blockchain. These malicious nodes are the ones that illegally attempt to break the trusted consensus mechanism, manipulating information about transactions, cause congestion of the network, and interruption of daily network activity. The blockchain system can thus become insecure, unstable and inefficient.

To solve problems mentioned above, we present an effective consensus algorithm DDPOS to limit the energy utilization and enhance consensus effectiveness of the blockchain network.

1.3 Motivation

- Crucial component of any blockchain system is the consensus algorithm, which in many ways determines its performance and security.
- The existing studies on consensus algorithm have incomplete discussions on the properties of the algorithm and fail to analyze their scope.
- Hence we develop a consensus algorithm to fill this gap by analyzing wide range of consensus algorithm.

1.4 Objective of the project

The objectives are as follows:

- To develop an algorithm which is faster than PoW and PoS.
- To improve efficiency of the transaction.
- To reduce power consumption.

1.5 Problem Statement

Any node might try to add corrupt block into a Blockchain. These nodes are nothing but called as malicious node. These malicious nodes secretly breach the established consensus process, modify with transaction details, causing congestion of the network, and disrupts normal network functionality. The blockchain scheme can thus becomes unsafe, unstable and inadequate.

We aim to quantify these issues by proposing an enhanced consensus that sets out PoW's principle to reinforce equity, and PoS's principle to limit the energy utilization and enhance consensus effectiveness of the blockchain network. Hence our problem statement is defined as: To develop a secure and efficient consensus algorithm for blockchain networks.

In the next chapter, we will discuss about the requirement analysis.

Chapter 2

REQUIREMENT ANALYSIS

In this chapter, we will address the system model, phases involved in requirement analysis and how the requirements' quality can be improved. We will also learn about the identified functional and non-functional requirements, and hardware and software specifications of the given project.

Requirements analysis includes activities that satisfy the needs and conditions for introducing a new or enhanced product or project, taking into account the possible conflicting objectives of the different stake-holders, evaluating, recording, validating and addressing specific needs. Requirements analysis is critical for a system or software project to succeed or fail.

Requirement analysis is of three types:

- Eliciting requirement:- recording of the business process, and stake-holder interviews. It is also known as requirement gathering or requirement discovery.
- Analyzing requirement:- determine whether the specifications specified are simple, complete, consistent and unambiguous and resolve any obvious conflicts.
- Recording requirement:- recorded in the form of a policy guidelines which can include natural language documents, use cases, user stories, process requirements and a range of models including data models.

Requirement Analysis can be a long and agonizing process involving many delicate psychological skills. Analyst will use a combination of different methods to determine stakeholders' exact requirements in order to produce a system that meets business needs. Requirements quality can be improved through the methods like:

- Visualization: the use of gadgets to better understand the desired end-product such as visualization and simulation.
- Consistent: producing a consistent set of models and templates to record the requirement.
- Documenting dependencies: recording dependencies and interrelations between requirement, and any expectation and congregation

2.1 System Model

In order to understand how to use models to construct system and conceptualize it, we draw system model. A system model is used to define and express different sources such as requirement analysis, model, development, action, input variables and output variables. The system model of our proposed consensus algorithm is shown in Figure 2.1.

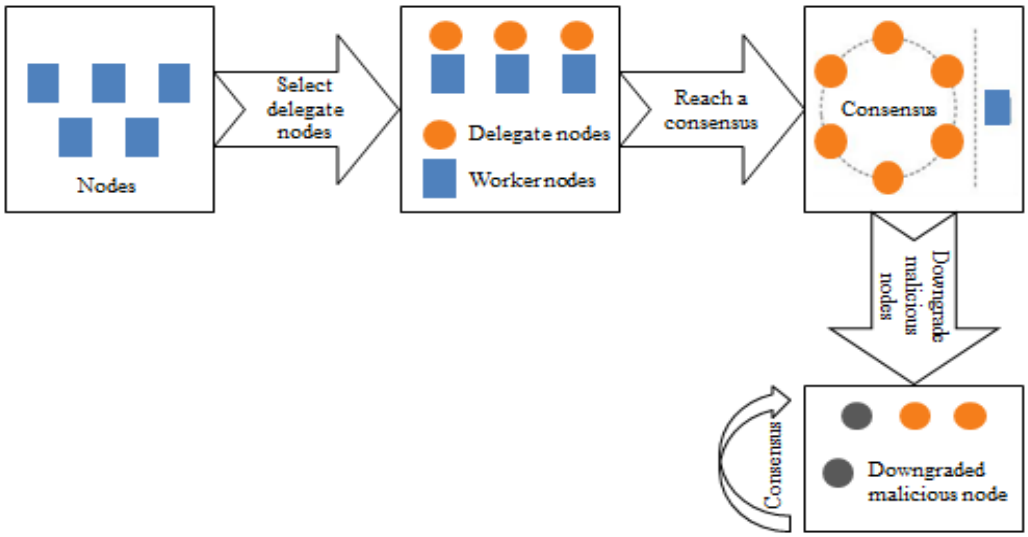

Figure 2.1: System Model

The DDPoS algorithm consensus process is composed of three phases:

i Selection of delegate nodes: every peer in a network are categorised to execute different roles to specific types of peers, and are mainly partitioned into two types: delegate nodes and non-delegate nodes. The key task of the non-delegate peer is to produce the transaction while the delegate peer is to produce and validate the data blocks.

ii Achieving consensus: this is achieved by delegate nodes. They contribute in reviewing the entire process from the production to the validation of blocks.

iii Downgrading malicious nodes: quick downgrade of malicious nodes to maintain good system functioning and safety.

2.2 Functional Requirements

Specifications of functions specify how everything needs to be achieved by defining its function, activity as well as operation needed to be accomplished. Analysis of functional specifications will be used as the functional analysis of the top-level functions.

The functional requirements as identified in the project given are as follows:

- The system should be able to implement PoW and PoS consensus algorithms.
- The system should be abli to implement DPoS algorithm.
- The system should detect and reduce the malicious nodes in the Blockchain.

2.3 Non Functional Requirements

Non functional requirements (NFR) sets out basis for evaluating a system's performance instead of associated factors. These are also known as "quality attributes" of a system. NFR are sometimes called as "qualities", "qualiy goals", "quality of service requirements", "constraints", "non-behavioral requirements", and "technical requirements", "ilities", from attributes like stability and portability. Further these requirements can be partitioned into two categories:

- Execution qualities:- which are observable during operation (at run time) such as safety, security and usability.
- Evolution qualities:- which are embodied in the static structure of the system such as testability, maintainability, extensibility and scalability.

The non-functional requirements as identified in the project given are as follows:

- The system should reduce the time by 5% to add the blocks to the Blockchain.
- The system should reduce the resource consumption by considerable amount.

2.4 Software and Hardware Requirements

Software Requirements:

- Programming Language: Python
- VMware Workstation
- Ubuntu 16.04 LTS
- Python 3.6
- Postman
- anaconda3 v5.2.0
- jupyter notebook v4.40
- cryptography, numpy and certain libraries

Hardware Requirements:

- Minimum 8GB RAM
- 512GB hard disk
- Minimum core i3 processor
- Minimum 5 peer nodes in VMware
- Minimum 2 delegate nodes

In the next chapter, we will be seeing about system design.

Chapter 3

SYSTEM DESIGN

In this chapter, we address the system design and how it proves to be beneficial during implementation phase. This section also describes about the architecture diagram and the flowcharts of consensus algorithms involved in the present work.

3.1 Architecture Design

System developers require diagrams to understand system architecture, explain and convey ideologies regarding the system design and requirement specification that the system needs to meet. This is a strong component which can be used during the planning process of the program to help partners understand the architecture, research agendas and clearly articulate motives. The architecture diagram of the proposed model is shown in Figure 3.1.

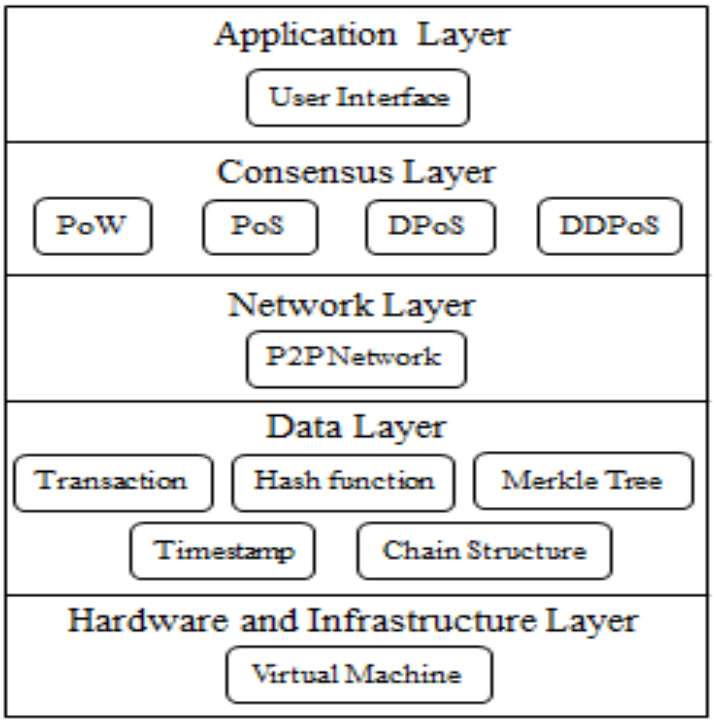

Figure 3.1: Architecture diagram of the proposed algorithm

Layered architecture has been adapted to implement the blockchain. All the peers are connected to same network to form a P2P network. The peers in the network make use of User Interface to communicate within the blockchain network. The consensus layer helps in reaching a consensus by computing the nonce value. The data layer comprises of hash value (previous block), transaction data, etc. These data are sent over the network for updating the ledger and synching the blockchain. The hardware and infrastructure is involved in running the program associated.

3.2 Flow Diagrams

A flowdiagram, or flowchart is a form of activity diagram that conveys a series of movements or progressions inside a model and represents a workflow of that system. Connectors and symbols function together to explain flow paths, moving objects and the quantity involved. It demonstrates the sequence of steps with proper design and construction in a quite effective and productive manner.

Flowchart has different shapes and symbols which are also called as flowchart symbols. In this case, we have used four types of shapes:

- The ones with rounded ends represent process starting and ending points
- The rectangles are used to signify intermediate steps or processes
- A diamond indicates a decision that can be answered in binary format (yes or no)
- An arrow is a connector that shows relationships between two shapes

There are different varieties of flowcharts, few of them are:

- Data flow diagrams
- Work flow diagrams
- Process flow diagrams
- Yes/no flow diagrams
- Decision flow diagrams

Flowcharts are originally designed to model the process flow for manufacturing companies. Today, flowcharts are used in the manufacturing, architecture, engineering, business, technology, education, research, medicine, government, administration and many other disciplines for a variety of purposes.

In order to implement DDPOS consensus algorithm, we implemented POW, POS and DPOS consensus algorithms. To understand how these algorithms work, we draw the flowchart of these four consensus algorithms.

Flowchart of PoW consensus algorithm:

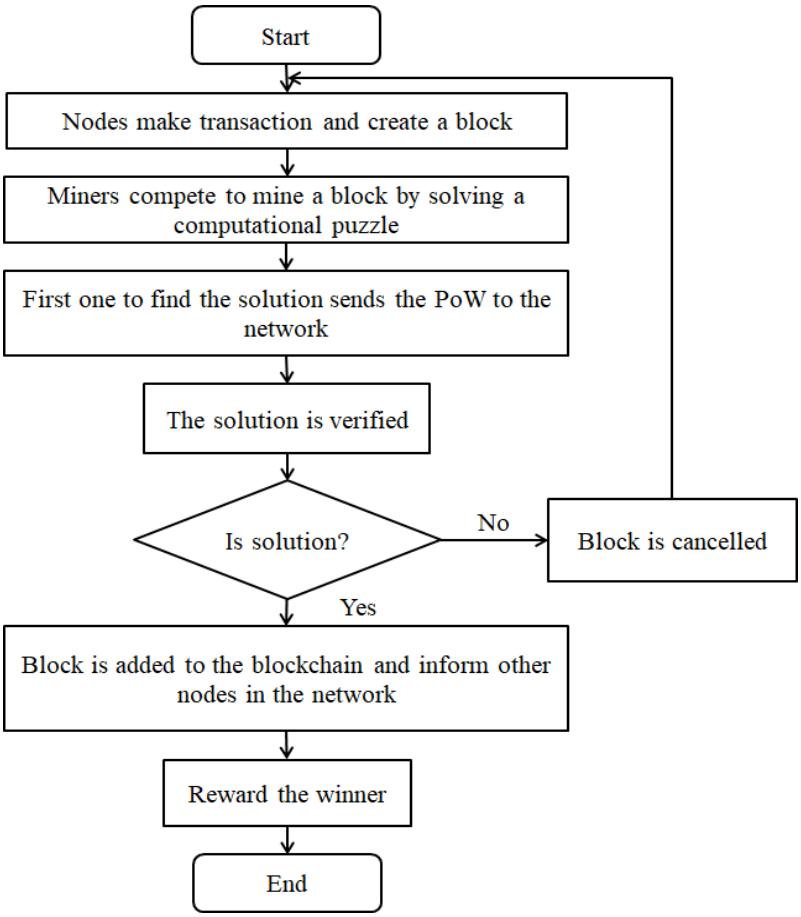

Figure 3.2: Flowchart of PoW algorithm

We present the brief explanation of PoW implementation. In PoW, the peers in network always want to write a decent block and receive reward by making a transaction to mine a block. The system requires to find out how to make the participant nodes that want to earn reward, "work" so the reward isn't discharged too easily. It is achieved by adding several combinations of alpha-numeric characters to the participants until the resulting hash includes a particular number of leading zeros.Therefore the nodes start competing to solve this computing puzzle. The first one who finds the solution sends his proof of work to the network. Once the solution is verified, the block created by the winning node, is appended to the blockchain. If the solution is wrong the block is created by it is cancelled. After mining a block, miner informs peers by sending an inventory message. If the peer is miner, after receiving the block can start mining his block. The miner then receives reward for his work. We can dynamically change the requirements (many more leading 0's) to ensure that the necessary work isn't too simple or too hard. This is called as adjusting the difficulty level. By raising the difficulty we can make the Proof of Work harder.

Flowchart of PoS consensus algorithm:

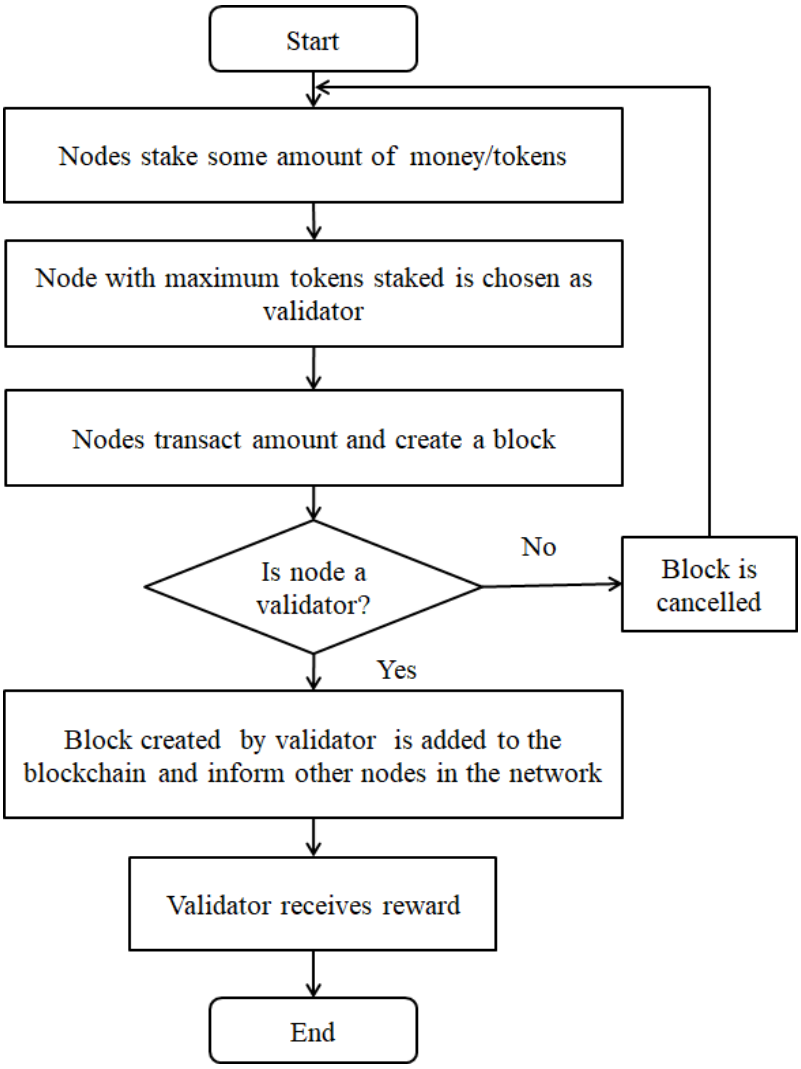

Figure 3.3: Flowchart of PoS algorithm

In Proof of Stake, instead of nodes competing with each other to solve computational hashes, blocks are minted or created depending on the amount of tokens each node is able to stake. Node with maximum tokens staked is chosen as "validator" or "leader". The tokens are Blockchain specific. Since nodes are "staking" their tokens, this consensus mechanism is called Proof of Stake. Then the nodes make transaction and create a block. The nodes are verified whether they are validators, if the node is not a validator the block created by it is cancelled and it cannot participate in the further transaction. The block created by leader after verification is added to the blockchain. The leader then informs peers by sending an inventory message. The validator or leader then receives reward. Since there are no forging devices in the network, only verifying machines taht verify new blocks is present, PoS can be regarded as low-cost, as the computational power power is limited.

Flowchart of DPoS consensus algorithm:

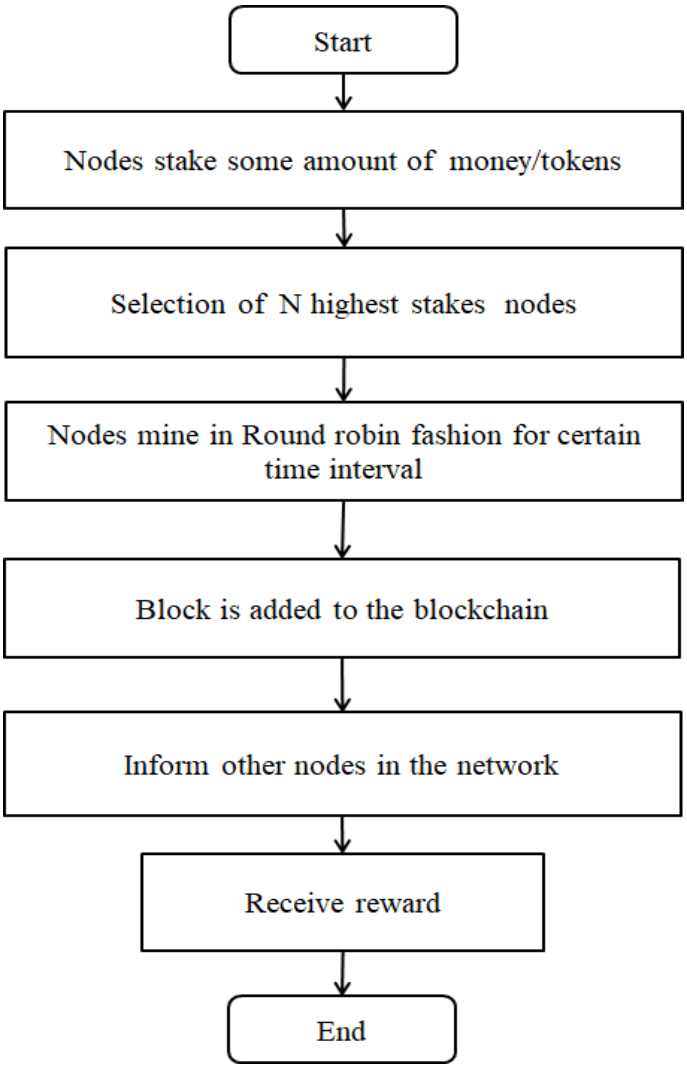

Figure 3.4: Flowchart of DPoS algorithm

DPoS is an improved consensus algorithm of PoS. Nodes in the network stake some amount of tokens inorder to mine a block. Nodes with maximum tokens staked are chosen as "validators" or "leaders". Among these validators, the top N nodes with highest staked tokens are chosen who can take turn in generating block. These nodes are called as delegate nodes and other nodes as non-delegate nodes. Peers make transaction and create a block. Only selected delegate nodes can mine a block in round robin fashion for certain time interval. If the delegate tries to mine block without its turn then its transaction is cancelled and a message is displayed saying "Not your turn. Hence cannot mine a block". After mining block in their respecting turns, block created by those delegates is added to the blockchain. Later delegates inform other peers in the network and sync the mined block. He then receives reward for mining. DPoS consumes low-cost compared to PoW and is more secure than PoS.

Flowchart of DDPoS consensus algorithm:

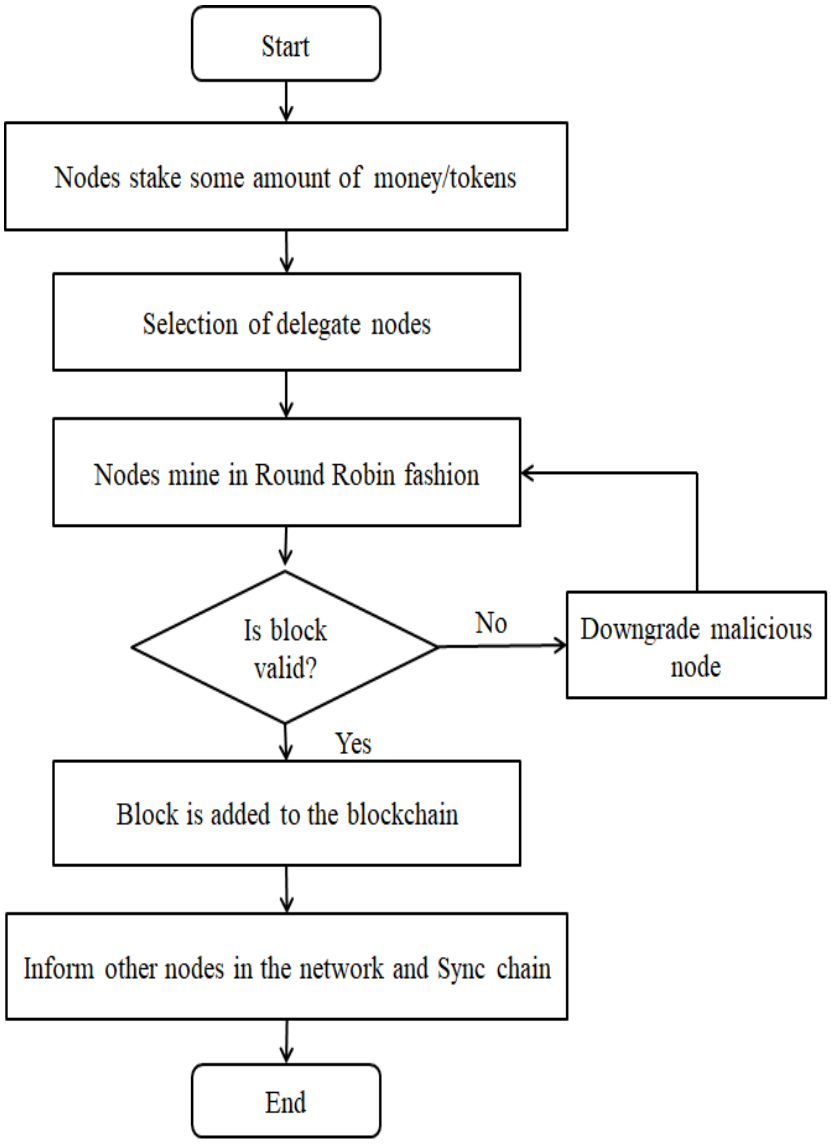

Figure 3.5: Flowchart of DDPoS algorithm

DDPoS is an improved consensus algorithm of DPoS. It works same as DPoS. Nodes in the network stake some amount of tokens inorder to mine a block. Nodes with maximum tokens staked are chosen as "validators" or "leaders". Among these validators, the top N nodes with highest staked tokens are chosen who can take turn in generating block. These nodes are called as delegate nodes and other nodes as non-delegate nodes. Peers make transaction and create a block. Only selected delegate nodes can mine a block in round robin fashion for certain time interval. If the delegate tries to mine block without its turn then its transaction is cancelled and a message is displayed saying "Not your turn. Hence cannot mine a block".

If the leader node among the highest staked nodes tries to add an invalid (corrupt) block to the blockchain, downgrading of such malicious nodes is done. Some amount of punishment fee is deducted from its account. Hence this algorithm is more secure then any other consensus algorithms implemented. After mining block in their respecting turns, block created by those delegates is added to the blockchain. Later delegates inform other peers in the network and sync the mined block.

In the next chapter, we will discuss about the implementation.

Chapter 4

IMPLEMENTATION

This chapter includes a brief outline of the algorithms used for the implementation of each module. The following section describes the algorithms used and the table 4.1 gives the definition of the symbols used in the following algorithms.

Table 4.1: Definition of symbols in the Algorithms

Symbol	Meaning
N^D	Delegate nodes list
V^N	Set of votes
N_Y^D	Faulty delegate node
$BLOCK_{ERROR}$	Corrupt block
m	Number of delegate nodes
n	Total number of nodes
A	Balance amount present in block
P	Punishment fee

4.1 Selection of Delegates

Algorithm 1 shows the implementation details for selecting delegate nodes among all nodes.

Algorithm 1 Selection of Delegate Nodes

Input: <N, stakes>

Output: N^D

broadcast <N, stakes>;
while $i <= n$ **do**
    N_i vote for N^D;
    $V_i^N <-$ count(N_i);
end while
quicksort(V^N); [in descending order]
$N^D < -N_i$ ($i \in$[1,m]);

4.2 Generating Block

Algorithm 2 shows the implementation details for generating block by delegate nodes.

Algorithm 2 Generating Block

Input: N^D, current_transactions[]

Output: new_block

initialize i to 0;
while $i <= m$ **do**
    if $i == m$ **then**
        re-initialize i to 0;
    end if
    N_i^D : mine_block(last_block, current_transactions[]);
    increment i;
end while

4.3 Downgrade Malicious Nodes

Algorithm 3 shows the implementation details for downgrading malicious node.

Algorithm 3 Downgrading malicious nodes

Input: $<BLOCK_{ERROR}$, stakes$>$

Output: msg

N_Y^D broadcast $<BLOCK_{ERROR}$, stakes$>$;
$N_{ERROR}^D < -N_Y^D$;
$N_Y^D[A] < -N_Y^D[A] - P$;
msg("Downgraded malicious node");

Chapter 5

TESTING

In this chapter we discuss the testing of the implemented consensus algorithm. Testing is one of the phases in the lifecycle of software development. Developers would then consider out that if their application and coding works within the test phase according to the requirements. And while all the failures that are found during the testing can not be solved, the outcome of this cycle can then be used to lower the amount of Software program errors. The primary aim of the Test Phase is to decide whether the proposed project is ready to deploy. The following section presents the acceptance testing on the proposed DPoS and DDPoS consensus algorithm.

5.1 Acceptance Testing

Acceptance Testing is a method of customer review for the validation/ admission of the software application, before incorporating the software application into the production level. It is done after the functional, integration and system testing is done in the last step of testing. The acceptance testing on the implemented DPoS and DDPoS consensus algorithms is shown in the following table 5.1 and table 5.2:

Table 5.1: Acceptance testing on DPoS consensus algorithm

Test ID	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	HTTP request to get balance	Display the current balance of the node.	Display the current balance of the node.
2.	HTTP request to get blockchain	List the current blockchain in the network.	List the current blockchain in the network.

3.	HTTP request to post transact
	{
	“recipient”: “123abgdr”,
	“amount”: 100
	}
		Transact the amount to the given recipient address.	Transact the amount to the given recipient address.

4.	HTTP request to post transact
	{
	“recipient”:,
	“amount”:
	}
		Internal server Error.	Internal server Error.

5.	HTTP post request to add Stake
	{
	“amount”: 100
	}
		Add the amount to the stake of the current node.	Add the amount to the stake of the current node.

6.	HTTP post request to add Stake
	{
	“amount”:
	}
		Internal server Error.	Internal server Error.

7.	HTTP get request to check Leader	Display the node with the highest stake among other nodes in the network.	Display the node with the highest stake among other nodes in the network.
8.	HTTP get request to check Stake	List the nodes and their current stake value	List the nodes and their current stake value.
9.	HTTP get request to mine (Leader node)	Mine a block by taking transaction from transaction pool	Mine a block by taking transaction from transaction pool.
10.	HTTP get request to mine (Non-Leader node)	Cannot mine because not leader	Cannot mine because not leader.

Table 5.2: Acceptance testing on DDPoS consensus algorithm

Test ID	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	HTTP request to get balance	Display the current balance of the node.	Display the current balance of the node.
2.	HTTP request to get blockchain	List the current blockchain in the network.	List the current blockchain in the network.

3.	HTTP request to post transact
	{
	“recipient”: “542abedh”,
	“amount”: 100
	}
		Transact the amount to the given recipient address.	Transact the amount to the given recipient address.

4.	HTTP request to post transact
	{
	“recipient”:,
	“amount”:
	}
		Internal server Error.	Internal server Error.

5.	HTTP post request to add Stake
	{
	“amount”: 100
	}
		Add the amount to the stake of the current node.	Add the amount to the stake of the current node.

6.	HTTP post request to add Stake
	{
	“amount”:
	}
		Internal server Error.	Internal server Error.

7.	HTTP get request to check Leader	Display the node with the highest stake among other nodes in the network.	Display the node with the highest stake among other nodes in the network.
8.	HTTP get request to check Stake	List the nodes and their current stake value	List the nodes and their current stake value.
9.	HTTP get request to mine (Leader node)	Mine a block by taking transaction from transaction pool	Mine a block by taking transaction from transaction pool.
10.	HTTP get request to mine (Non-Leader node)	Cannot mine because not leader	Cannot mine because not leader.
11.	Leader node corrupting the last hash of the block	Display: Cannot replace. The incoming chain is invalid	Display: Cannot replace. The incoming chain is invalid.
12.	Leader node syncing the corrupt block	Punishment fee will be applied to the node	Punishment fee will be applied to the node

Chapter 6

RESULTS AND DISCUSSION

This chapter gives a brief description about the details of the experimental setup to validate the proposed model's sustainability, and results analysis.

6.1 Simulation Setup

We have implemented POW, POS, DPOS and DDPOS consensus algorithms. The presented algorithms are implemented using Python language. In this experimental setup, an environment of five nodes in the network has been used, where two nodes are delegate nodes. We are able to add stake with stakes in the network, make transactions and mine the block. While mining the block it is confirmed that the request has come from a delegate node otherwise the request is declined. The results of the consensus algorithm implemented are shown in the following figures.

Results of DDPoS algorithm:

```
incoming connection from ('192.168.61.6', 37156)
added 192.168.61.6
incoming connection from ('192.168.61.6', 37158)
incoming connection from ('192.168.61.6', 37160)
incoming connection from ('192.168.61.6', 37162)
incoming connection from ('192.168.61.5', 53846)
Sent to 192.168.61.6
added 192.168.61.5
incoming connection from ('192.168.61.5', 53848)
incoming connection from ('192.168.61.5', 53850)
incoming connection from ('192.168.61.5', 53854)
incoming connection from ('192.168.61.7', 56752)
Sent to 192.168.61.6
Sent to 192.168.61.5
added 192.168.61.7
incoming connection from ('192.168.61.7', 56754)
incoming connection from ('192.168.61.7', 56756)
incoming connection from ('192.168.61.7', 56762)
incoming connection from ('192.168.61.8', 56504)
Sent to 192.168.61.7
Sent to 192.168.61.6
```

Figure 6.1: Nodes connecting to the network.

The Figure 6.1 shows the incoming connections from nodes to join the network.The peers in the network after connection are: 192.168.61.5, 192.168.61.6, 192.168.61.7 and 192.168.61.8.

```
client.validators
```

```
{'192.168.61.6': 60, '192.168.61.5': 30, '192.168.61.7': 90}
```

```
client.get_leader()
```

```
'192.168.61.7'
```

```
print(client.peers)
client.wallet.address
```

```
{'192.168.61.5', '192.168.61.6'}

'eee94217'
```

Figure 6.2: Peer displaying leader node.

The Figure 6.2 shows the output of the peer displaying all the nodes with their stakes and the leader node among them. The peer 192.168.61.7 is the leader with the highest stake of 90.

```
1 client.blockchain.to_json()
```

```
[{'timestamp': 1,
  'last_hash': 'genesis_last_hash',
  'hash': 'genesis_hash',
  'data': [],
  'difficulty': 3,
  'nonce': 'genesis_nonce'},
 {'timestamp': 1591521509.9766338,
  'last_hash': 'genesis_hash',
  'hash': '08b318e8bc6ea4f6deb30c2cb95886d807383f6d1ab84efc49bf1
a2167c48a33',
  'data': [{'id': '0f3f67a4',
    'output': {'0': 90, 'eee94217': 840, 'f902f999': 70},
    'input': {'timestamp': 1591521504.8469806,
     'amount': 1000,
     'address': 'eee94217',
     'public_key': '-----BEGIN PUBLIC KEY-----\nMFYwEAYHKoZIzj0C
AQYFK4EEAAoDQgAEN9K43uzG8w+CXPvMfOoml1HpmL0BPcLJ\nVNXUWdlGAp0IUG
```

Figure 6.3: Blockchain before corrupting.

In Figure 6.3, the peer is displaying the blockchain before corrupting that is before mining invalid block.

```
client.blockchain.to_json()[1]["last_hash"] = "corrupt"
```

```
client.blockchain.to_json()
```

```
[{'timestamp': 1,
  'last_hash': 'genesis_last_hash',
  'hash': 'genesis_hash',
  'data': [],
  'difficulty': 3,
  'nonce': 'genesis_nonce'},
 {'timestamp': 1591521509.9766338,
  'last_hash': 'corrupt',
  'hash': '08b318e8bc6ea4f6deb30c2cb95886d807383f6d1ab84efc49bf1
a2167c48a33',
  'data': [{'id': '0f3f67a4',
    'output': {'0': 90, 'eee94217': 840, 'f902f999': 70},
    'input': {'timestamp': 1591521504.8469806,
     'amount': 1000,
     'address': 'eee94217',
     'public_key': '-----BEGIN PUBLIC KEY-----\nMFYwEAYHKoZIzj0C
```


Figure 6.4: Blockchain after corrupting.

In Figure 6.4, the peer is displaying the blockchain structure after corrupting. The block is corrupted by changing the previous hash value of the block. It is seen from the structure that the last_hash value is changed from "genesis_hash" to "corrupt".

```
client.blockchain.to_json()
```

```
[{'timestamp': 1,
  'last_hash': 'genesis_last_hash',
  'hash': 'genesis_hash',
  'data': [],
  'difficulty': 3,
  'nonce': 'genesis_nonce'}]

Adding trasaction
Recieved
Done
Adding trasaction
Recieved
Done
Cannot replace. The incoming chain is invalid: The block last_ha
sh must be correct
Cannot replace. The incoming chain is invalid: The block last_ha
sh must be correct
```


Figure 6.5: Message after the corrupted block is synced.

In Figure 6.5, Peer is displaying the message after the corrupted block is synced. Peer identifies the malicious node and displays message: "Cannot replace. The incoming chain is invalid: The block last_hash must be correct." after the block is synced to the blockchain.

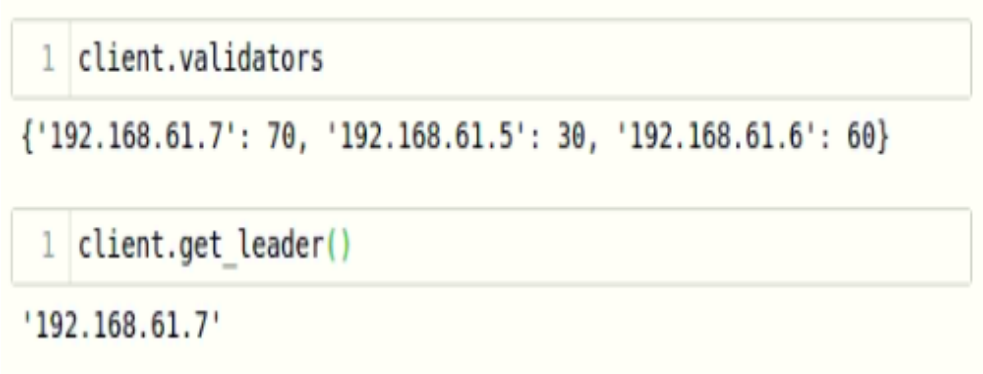

Figure 6.6: Deduction of stake after corrupting the block.

In Figure 6.6, Peer shows the deduction of stake from the balance amount, as the Punishment fee because of corrupting the block.

6.2 Result Analysis

After implementation we carried out performance analysis on these algorithms based on their difficulty level, execution time etc.

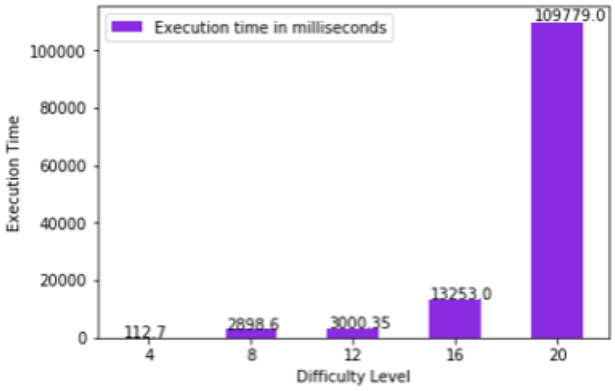

Figure 6.7: Performance analysis based on Difficulty level.

The Figure 6.7 shows the variation of execution time with increasing difficulty level. It can be seen that as the difficulty level goes on increasing the execution time increases. This is because the difficulty level is proportional to time taken to mine the block.

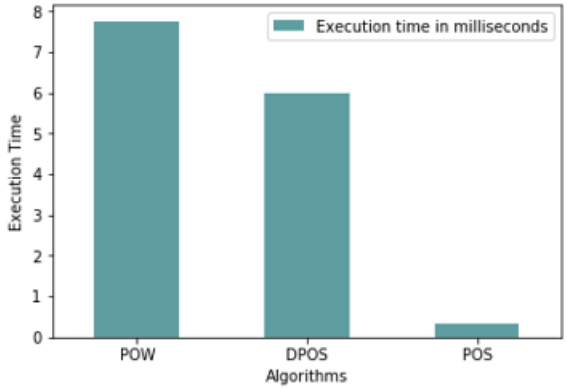

Figure 6.8: Performance analysis based on Execution time.

The Figure 6.8 shows the execution time of these algorithms in milliseconds. It can be seen that PoW takes more time than the other algorithms as all the nodes compete with each other to mine the block whereas PoS is fastest as a single node having the highest stake mines the block in this algorithm. DPoS takes moderate time for the execution as it has to even elect the delegates who mine the block.

Chapter 7

CONCLUSION

For a node with sufficient resources in a blockchain network , it is necessary to get a fair chance to generate the block. By implementing a selection mechanism that gives a list of delegates, we have achieved fairness in the system. Also the system is developed with two sets of nodes i.e. delegate nodes and non-delegate nodes so as to avoid maximum utilization of resources. The delegate nodes are forced to deposit their stakes to avoid any malpractices in the blockchain environment. If any malicious node is found in the network its stakes are reduced to zero, this ensures security in the system.

We have implemented an efficient consensus algorithm called DDPOS for lower resource usage, greater operational efficiencies and greater blockchain security.

REFERENCES

[1] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. "Blockchain Technology Overview". Technical report, National Institute of Standards and Technology Internal Report 8202, U.S. Department of Commerce, 2018.

[2] Mahendra Kumar Shrivas and Dr. Thomas Yeboah. "The Disruptive Blockchain: Types, Platforms and Applications". 5th Texila World Conference for Scholars (TWCS), on Transformation: The Creative Potential of Interdisciplinary Multidisciplinary Knowledge Exchange, 12 2018.

[3] Y. Yuan, F. Y. Wang. "Towards Blockchain-based Intelligent Transportation Systems". pages 2663–2668. IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1-4th November 2016.

[4] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". 2008. Available online: `https://bitcoin.org/bitcoin.pdf`.

[5] Fahad Saleh. "Blockchain Without Waste: Proof-of-Stake". *SSRN Electronic Journal*, 9th July 2018. Available: `http://dx.doi.org/10.2139/ssrn.3183935`.

[6] Dantheman. "DPOS Consensus Algorithm - The Missing White Paper". 2016. Available online: `https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper`.

[7] Miguel Castro and Barbara Liskov. "Practical Byzantine Fault Tolerance". *Third Symposium on Operating Systems Design and Implementation, New Orleans, USA*, pages 173–186, February 1999. Available: `http://dx.doi.org/10.1145/571637.571640`.

Appendix A

Explanation of tools

A.1 VMware Workstation

VMware Workstation supports cloud computing and virtual machine software with x86 and x86-64 bit machines that operate various operating systems on one virtual host computer. Every other virtual machine can concurrently run a single instance of any operating system (Microsoft, Linux, etc).

A.2 Postman

Postman is an immersive, instantaneous tool for evaluating ones project's APIs. Developers can easily build, exchange, check and record the APIs. It is accomplished by allowing users to build and also save simple and complex HTTP requests and also to review ones response. The outcome is perhaps more effective and far less painstaking task. This software has become one of the free and open - source applications, which is essentially an extension to the browser.

A.3 Jupyter Notebook

Jupyter Notebook is indeed an open-source web application that enables users to build and share textual content, equations, visual analytics and narratives in real time. The applications encompasses processing and interpretation of data, computational analysis, numerical optimization, machine learning, data presentation and perhaps more.