Decentralized Currency & Document Time-Stamping System

Harsh Borkar
Electrical Engineering Department
Indian Institute of Technology Bombay
Mumbai, India
190070027@iitb.ac.in

Vipin Ochiramani
Electrical Engineering Department
Indian Institute of Technology Bombay
Mumbai, India
190070073@iitb.ac.in

Prasann Viswanathan
Electrical Engineering Department
Indian Institute of Technology Bombay
Mumbai, India
prasann@iitb.ac.in

***Abstract*—In this paper we propose a currency and document sharing system with a decentralized approach, inspired by the working of Bitcoin. Blockchain is used as the data structure behind the system. We describe components of the system, protocols of functioning and the working principle. We conclude with analysis on our methods and describe the novelty of work. Also, we briefly describe the implementation of the system in real life.**

***Index Terms*—blockchain, file sharing, crypto-currency**

I. Introduction

Most of the services we use are centralized and require a central authority. They can easily be developed and deployed. But there are some disadvantages of centralized systems:-

1) These systems don't provide transparency to the users.
2) Data once mutated cannot be recovered back.
3) The server node becomes a bottleneck and if it gets compromised or broken, user data becomes vulnerable.
4) Scalability of the system is difficult

Blockchain is the solution to all the above problems. Blockchain is a distributed ledger that eliminates the need of a central authority and allows nodes to communicate directly with one another. Transactions in blockchain are recorded on blocks in a shared ledger available to all the nodes in the network. Since the ledger is shared, anyone and everyone can see all the transactions that occur since the start of blockchain.

Blockchains are very inefficient, because we are asking every node to store and repeat the same work. But this inefficiency provides blockchain immutability and trust, by asking most of the nodes to agree on the common truth.

Blocks in the blockchain are all chained together starting from the first block. Chaining combined with hashing provides the blockchain immutability, which means there is no possibility of altering the data present in the blocks [1].

In this paper we propose methods to improvise the bitcoin network. We also aim to build a time-stamping system for files, which will help to verify the authenticity of file shared along with the timing.

II. Implementation

A. Blockchain structure

The proposed blockchain's structure consists of :

- Nonce (256 bits)
- File Hash (256 bits)
- Transaction Hash (256 bits)
- Previous block's Hash (256 bits)
- Timestamp (64bit)
- Transaction Summary block Hash (256 bits)
- Additional Data field (1024 bits)

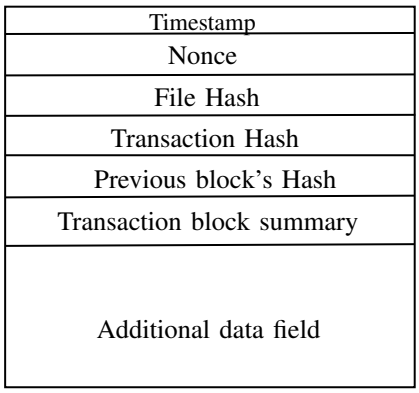

Fig. 1. Blockchain Structure

1) Nonce: The nonce is a 256-bit unsigned integer which is used to satisfy the proof of work rule (later stated) [1]

2) File Hash: File hash is the top node of the binary tree containing all the hashes of files associated with this block. It is a 256 bit integer calculated using SHA256 Hashing algorithm.

3) Transaction Hash: This data block is similar to file hash. Transaction hash is the top node of the binary tree containing all the hashes of transactions associated with this block. Transaction hash is a 256 bit integer calculated using SHA256 Hashing algorithm.

4) Previous block's hash: Previous block's hash contains the hash of the previous block in the blockchain system. [1]

5) Timestamp: The timestamp is a 64bit UNIX timestamp denoting the time at which the solving for proof-of-work algorithm had started.

6) Transaction Summary Block Hash: Transaction summary block is another data structure that is used for keeping a record of all transactions in the block. Transaction-Summary-

Block Hash is a 256-bit hash of transaction summary block calculated using SHA256 algorithm.

7) Additional Data Field: Additional Data Field block is a data field of size 1024 bits where any data can be placed.

B. Users

For a person to interact with the blockchain network, they must have a user account. The user account is defined by a unique 1024 bit RSA public key. The public key of the account is the identity of the account. The 1024 bit RSA private key is kept secret and is used for signing transactions as defined later in this paper.

There is also a special type of account called as sink account. The special property of this account is that coins can only be deposited to this account and can never be reused for transactions again.

C. Transactions

Transaction is defined as a request to add a file hash to a block or a request to transfer coins from one account to another. Each transaction has transaction fees, which are paid to the miner who mines the block.

Transactions are of two types:

- Coin Transactions
- File Transactions

1) Coin transactions: These type of transactions denote the amount of coins to from payer to payee's account. This coin transaction must be signed by payer's private key for successful transaction request. Coin transactions are subject to transaction tax, defined later in this paper.

2) File transactions: These transactions include a hash of a file which needs to be added to a block. The transaction must be signed by the private key of the user requesting to add the file to the blockchain.

3) Transaction summary block: Transaction summary block describes all the transactions of the current block merged into a single block.

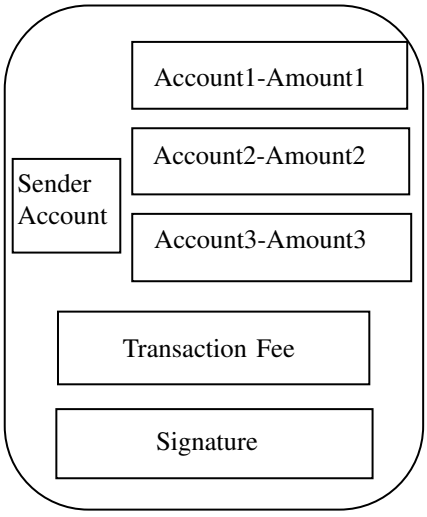

Fig. 2. Coin Transaction Block

D. Network Hierarchy

We define a *node* to be any participant in the blockchain activity. In a network following types of *node*s are present:

1) Passive Node: Passive nodes are the types of nodes who do not participate in mining blocks but add transactions to transaction pools and files to file pools.

2) Active Node: Active nodes perform all the functions of Passive nodes and also take part in mining blocks.

3) Sub-Head Node: Sub-Head nodes perform all the functions of Active nodes. These are the nodes which are contacted by Active and Passive nodes for broadcasting transactions, file-hashes, mined blocks, etc. They also perform verification of newly generated blocks.

4) Head Node: Head node performs all the functions of Sub-Head node and also issues certificates to Sub-Head nodes for Sub-Head node authentication. The IP-address of the Head Node is known to everyone in the network.

III. Protocols

All these operate on top of application layer protocols. These protocols work with two messages only, both for end-point validation and data-transfer. Two messages are enough for this use-case because data sent need not be sent in encrypted form because the system itself is decentralized.

A. State of Chain

To get the current state of chain, a node should contact Sub-Head or Head nodes. The procedure is as follows:

1) Node sends a GET request to Sub-head/Head Node.
2) Sub-Head/Head replies with current state of blockchain and a randomly generated nonce. This message is signed with the *status key*. The Sub-head nodes also send their certificate issued by the Head node along with these.

The *status key* is generated by hashing Sub-Head/ Head Node's private key appending with the randomly generated nonce [2].

B. Transaction Pool

To get all the transactions which have not yet been added in the blockchain

1) Node sends a GET request to Sub-head/Head node.
2) Sub-Head/Head replies with all pending transactions and a randomly generated nonce. This message is signed with the *status key*. The Sub-Head nodes also send their certificate issued by the Head node.

The *status key* is generated by hashing Sub-Head/Head node's private key appended with the randomly generated nonce.

C. Broadcast Transaction

Any newly generated transaction should be sent to Sub-Head/ Head node for broadcasting it to the network.

1) Node sends the transaction Sub-Head/ Head node.

D. Broadcast of a new block

Any newly generated block is sent to Sub-Head / Head Node for broadcasting it to the network.

1) Node sends the new block to the Sub-Head/ Head node.
2) Sub-Head/Head node responds with a Boolean value whether the given block was accepted or not. This message is signed with the *status key*. The Sub-Head nodes send their certificate issued by the Head Node.

The *status key* is generated by hashing Sub-Head/ Head Node's private key appending with the randomly generated nonce.

E. Synchronization of Sub-Head nodes and Head node

As soon as a new transaction is received by a Sub-head/Head node, it is broadcasted to all current Sub-Head nodes and Head, using the same procedures described before.

F. Issuing Certificates to Sub-Head Nodes

Head node issues certificates to the sub-head nodes—these certificates expire after a specified number of blocks. The head-node also keeps a certificate revocation list which contains all the blocked certificates.

IV. Peer Networking

A. Delegated Certification Authorities Model

The proposed systems uses a Delegated Certification Authorities model. The *Head Node* acts as the root Certificate Authority, where as, *Sub-Head Node* acts as the Delegated Certificate Authority.
The Delegated Certificate Authorities are issued certificates which are valid for a limited time. Such scheme is feasible where decentralization is needed inside a particular organizational group. For completely decentralized model, certificates with infinite time validity can be issued.
For the completely decentralized model, there will be a root certificate authority for completeness. In addition, the *Sub-Head* nodes will have the authority to issue certificates to any node. Hence, each node will have can become *Sub-Head* node capable of all the actions of the *Head Node*.

B. Locating Peers

Each node has the address of the *Head Node* and a self-signed certificate embedded into it. Each node may communicate with any of the *Sub-Head* node or *Head Node*.
Each node keeps a list of *Sub-Head* nodes (in the form of collection of certificates) along with a confidence level for each node. For every successful or incorrect communication the confidence level increases or decreases accordingly. While connecting to a *Sub-Head* node, the node tries one-by-one each *Sub-Head* node in its list of *Sub-Head* nodes, until it finds a node which is online.
Nodes may also share the certificates along with their confidence levels to other nodes in the same network.

V. Rules of Operation

A. Proof of Work

We define *difficulty of mining* as the number of zeros with which the block's hash should begin with. A block is considered to be generated correctly if it satisfies difficulty of mining for the given block [1].
The *difficulty of mining* is dynamic and may change from block to block. The rules are :

- If the time required to generate the latest block (found by comparing the last two timestamps) is lesser than 60 seconds, then the *difficulty of mining* of the current block is increased by 1.
- If the time required to generate the latest block is greater than 120 seconds, then the *difficulty of mining* of the current block is reduced by 1.
- In all other cases, the *difficulty of mining* is kept same.

B. Veracity of Blockchain

Define *computations stored* of a blockchain to be the sum of *difficulty of mining* of all blocks in the blockchain. Then, a blockchain with the greatest *computations stored* is considered as the most valid chain.

C. Transaction Tax

For every Coin Transaction, a small percentage of the coins transferred is removed as transaction tax. The transaction tax destroys those coins and the transaction tax is not transferred into any account.

D. Block Reward

For every new valid block generated, the node which generated the block is rewarded with a specified number of coins [1]. The number of coins rewarded is calculated as a function of the number of coins in the blockchain.

E. Coin Indivisibility

All the coins the coins in the network are indivisible, meaning that, the smallest unit of currency is one coin. After deduction of taxes, if the balance in the account is in fraction, then the coin balance is rounded-down to the nearest integer.

VI. Analysis

A. Block Reward

Let c denote the targeted coins in circulation, a denote the current coins in circulation, d_1 and d_2 are constants.
The aim is put $c = 10^9$ coins in circulation. The block reward for a current block is defined as follows :

$$\text{Block Reward} = \min(\frac{c-a}{d_1}, d_2) \tag{1}$$

Assuming one block every minute, the aim is that the supply of coins increases linearly for the first one year and then it follows an exponential decay.

$$\text{Coins mined in the first year} = 365 \times 24 \times 60 \approx 5 \times 10^5 \tag{2}$$

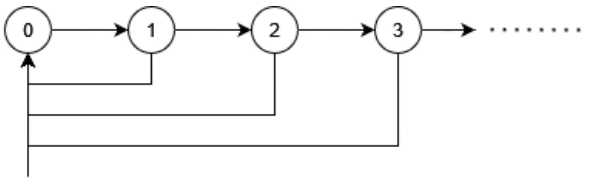

Fig. 3. State Transition Diagram of the Continuous Time Discrete Markov chain. The forward transition rate is λ and the reverse transition rate is $k\mu$.

As more and more blocks get mined, the a increases and for linear increase in coins,

$$\frac{c-a}{d_1} \geq d_2. \tag{3}$$

Thus, for a linear increase in the coins for the first year, substituting $a = 5 \times 10^5$ in (3), the satisfying values are $d_1 = 5 \times 10^5$ and $d_2 = 10^3$.

$$\text{Block reward} = \min(\frac{c-a}{5 \times 10^5}, 10^3) \tag{4}$$

B. Transaction Tax

Assume the transaction tax is t such that $t \ll 1$. Assume we have 1 coin. After every transaction the value of the coin becomes $(1-t)$ times. After n transactions the value becomes $(1-t)^n$ times. Calculating the half life of the coin,

$$0.5 = (1-t)^n \tag{5}$$

Thus, the half-life of the coin is $\log_{(1-t)} 0.5$.

C. Effect of Transaction Tax

As stated earlier, the minimum unit of currency is the coin itself, and the coin balances are rounded down to the nearest integer after each transaction.
Suppose, we have c coins with us and these coins are transferred round and round again. Then after n transactions, the coins remaining with us are,

$$\text{Coins remaining} = \lfloor c \cdot (1-t)^n \rfloor \tag{6}$$

Notice that the RHS of the equation becomes zero when,

$$c \cdot (1-t)^n < 1$$

. Thus, when c coins are given, the coins become zero in value after,

$$\text{Coin Life} \approx \log_{(1-t)}(c^{-1}) \tag{7}$$

D. Transactions pending in the system

We define a Continuous Time Markov Chain where the number of transactions pending in the system defines the state of the system. We assume that new transactions arrive as per a Poisson process of rate λ. Assume that each node which is mining is producing blocks as per a Poisson process of rate μ. Suppose there are k nodes which are mining in the system. Therefore the process of arrival of newly mined blocks is a Poisson process with a rate $k\mu$.
The state transition diagram for the Continuous Time Markov Chain is depicted in the figure 3. From the same figure, we get,

$$\pi_{n+1} = \frac{\lambda}{\lambda + k\mu}\pi_n \tag{8}$$

$$\sum_{n \geq 0} \pi_n = 1 \tag{9}$$

Here, π_i denotes the probability that the system has i transactions pending in the transaction pool. Solving the equations, we get,

$$\pi_n = \left(\frac{\lambda}{\lambda + k\mu}\right)^n \left(\frac{\lambda + k\mu}{k\mu}\right) \tag{10}$$

Thus, the expected number of transactions in the system is,

$$E[\text{Transactions}] = \frac{\lambda(\lambda + k\mu)^2}{k^3\mu^3} \tag{11}$$

The more the expected number of transactions the more will be the utilization of the computational power of the network. This value can be kept under control as the parameter μ is dependent on the $difficulty$ of the block mined.
This quantity is particularly important in determining the efficiency of the system. This is because, by PASTA (Poisson Arrivals See Time Averages) theorem, the expected number of transactions in a block is also the same as this quantity. We want to maximise the number of transactions in a block, but this comes at the cost of decreasing $k\mu$ (rate of arrival of new blocks).

E. Bandwidth Requirements and Data Storage

The size of a block is 2368 bits and the size of a transaction is 1088 bits (1024 bits account number and 64 bits amount to be transferred). Continuing with assumption of arrival rate of transactions to be a Poisson process with a parameter λ and that of blocks creation to be $k\mu$, the amount of data generated per second is given by,

$$\text{Data creation rate} = k\mu \cdot 2368 + \lambda \cdot 1088 \tag{12}$$

As long as this value is lesser than the available bandwidth of the medium, there would not be congestion and the blockchain would function correctly.
Focusing on the data generated by Blocks alone, assuming $k\mu = 1$ block/2 minutes, the rate of data generation is 19.733 bits/second. This means that around 6.2 Gigabits of data will be generated by the blocks alone. This figure is manageable with respect to the modern day storage devices.

VII. Novelty

In our blockchain, we are time-stamping the files and documents along with hashing it, so that we can verify the authenticity and timing of the files later if needed.
For coin-base we plan to add transaction tax on each transaction, by this we aim to control the number of coins in circulation, hence combating inflation. We also plan to vary the miner's reward in a different way compared to bitcoin, where initial miners got undue advantage. In our model new miners have status similar to old ones.

The first server node acts as Head node and issues certificates to nodes willing to act as Sub-Heads. This way we ensure that head node does not become the bottleneck.
The coins have a definite life time, that is, after a finite number of transactions the coins' value reduces to zero. This is due the rule of *Coin Indivisibility*. Thus, the we ensure a finite supply of coins in the system.
This system is can be easily converted into partially decentralized system, where in the Head Node is stringent when it comes to the distribution of certificates to the *Sub Head Nodes.* The system developed is flexible (due to the presence of the additional data field in the block chain)and more applications can be built upon it using this framework as a backbone.

VIII. Future Work

The Peer Networking algorithms is still slow as it runs through a list and trying each of nodes for connection requests. Also it does not take into account the latency or the cost of connections. The Peer Networking algorithms can be developed such that the networks of connections form an efficient minimal spanning tree.
The proposed idea of transaction tax and the variable block reward system needs to be researched more rigorously and should be checked for inconsistencies.

IX. Contribution

1) Harsh Borkar (190070027):- Decided the structure of blockchain, transaction methods, analysis of coins in network, and few protocols and rules of operation to be used in blockchain. Developed and tested the application for simulation of the proposed decentralized system.
2) Vipin Ochiramani (190070073):- Decided rules of operation, protocols to be used, analysed coin-base, and modified blockchain structure. Debugged the incorrect implementations of the system.
3) Prasann Viswanathan (190070047):- Worked on Network hierarchy, modified protocols, reward structure, and taxation. Designed the application data structure for storage.

More details about the application implementation can be found in Appendix section.

X. Appendix

This section discusses how the application was implemented in real life.
Our system runs on application level protocols and needs to communicate with other peers on the network. Hence, the system was implemented using Django Framework (based on Python programming language). Also, Django makes managing databases easy for beginners.
For peer networking and the other protocols, Hyper Text Transfer Protocol (HTTP) was used. HTTP was chosen because it enables transfer of data relatively easily without worrying about underlying lower level networking protocols.
For database, SQLite database was used.

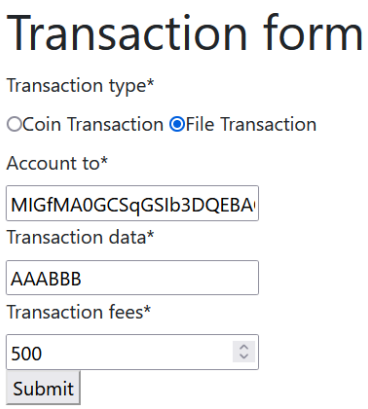

Fig. 4. Transaction form. Users can use this transaction form for adding coin transactions or file transactions into the transaction pool.

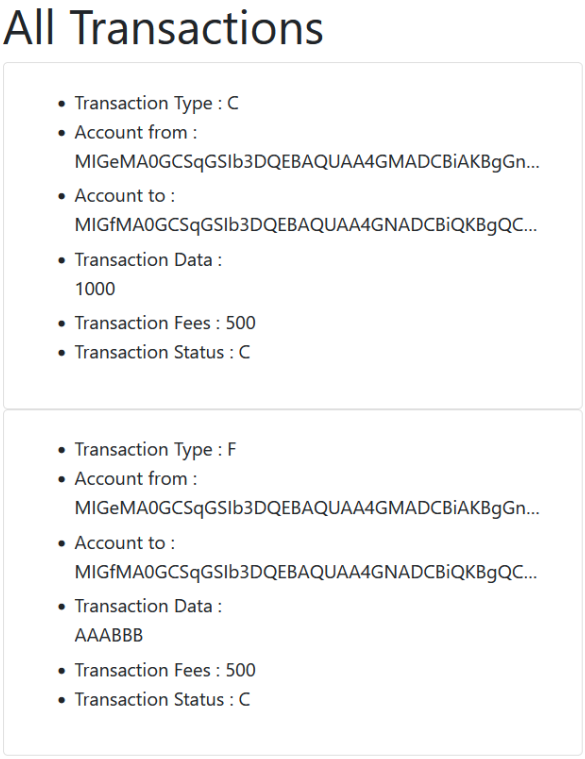

Fig. 5. A section for displaying all the transactions in the transaction pool.

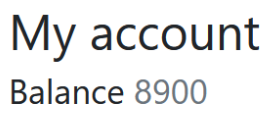

Fig. 6. A section for displaying the available balance of coins in one's account.

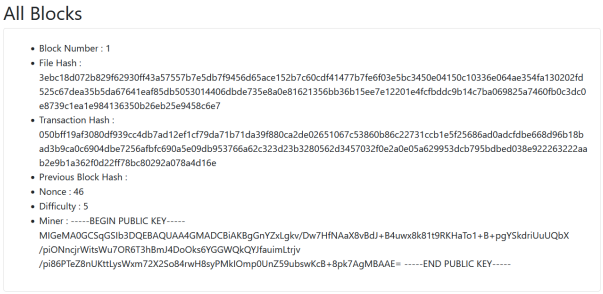

Fig. 7. A section for displaying the blocks currently in the block chain.

Figures 4, 5, 6 and 7 show some of the snapshots of the application developed.

Due to a lack of time, the implementation shown needs more work in terms of blocks supporting multiple transactions, more robust peer networking algorithms and the implementation of transaction summary block needs to be done.

REFERENCES

[1] "Bitcoin: A Peer-to-Peer Electronic Cash System," Jan. 30, 2014. Accessed on: March. 19, 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980