Blockchain: Overview, Practical Implementation & Its Uses

Shubham Kumar[1] , Sanchita Jaiswal[2]

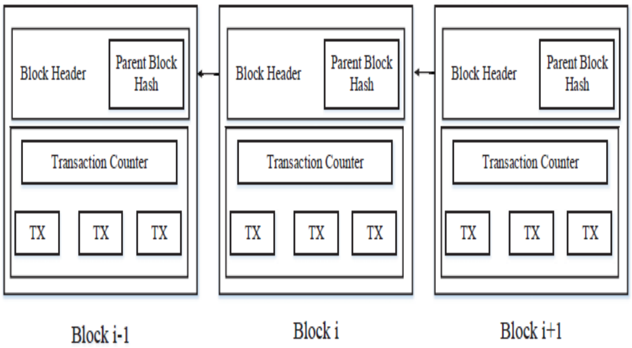
Assisstant Professor

[1]Department of Computer Science, MIET, Meerut

[2]Senior Executive Cyber Security, Tata Advanced Systems Limited,Noida.

Abstract

Advanced from the Merkle Tree, Blockchain Technology is a completely decentralized computerized register which keeps a protected history of information trades. The decentralization part of Blockchain Technology does away the need of any focal specialist for overseeing it. In this paper we present an exhaustive diagram on blockchain innovation. We initially start by revealing insight into the basics of Blockchain Technology then we investigate some run of the mill calculations utilized in different blockchains. Blockchain, the establishment of Bitcoin, has gotten broad consideration as of late. Being an ineradicable information putting away innovation, Blockchain can be utilized in budgetary resources as well as anything which has some worth. In any case, being a human development, drawbacks are even here in the blockchain innovation, for example, versatility issues, security issues, and not really easy to use for non-specialized individuals. Next, with regular specialized issues we have discussed the ongoing advances. We in conclusion finish up this paper by spreading out conceivable future improvements of blockchain innovation.



Key Words: Blockchain, Ledger, Bitcoin,Uses

INTRODUCTION

Blockchain likely could be seen as an open record and each submitted dealings is place during a rundown of squares. This chain creates as new squares are mounted to that perpetually. With a terribly planned information stockpiling structure, exchanges in Bitcoin framework may happen with no any outsider and hence the center advancement to develop Bitcoin is blockchain, that was starting arranged in 2008 and dead in 2009 [1]. Nowadays advanced money has turned into an a la mode articulation in each exchange and significant world. In show of the premier famous advanced money, Bitcoin has charmed a gigantic accomplishment with its capital market accomplishing ten billion greenbacks in 2016 [2]. Hilter kilter cryptography and conveyed accord figuring are dead for purchaser security and record consistency. The blockchain innovation has key characteristics of decentralization, steadiness, namelessness and auditability. With these properties, blockchain will extensively save the cost and upgrade the efficiency. As an issue of first significance blockchain is perpetual. Dealings can't be changed once it's full into the blockchain. Associations that need high duty and trustworthiness will use blockchain to attract buyers. Besides, blockchain is appropriated and may keep away from the main motivation behind disillusionment situation. Blockchain are regularly used in a few cash benefits for instance, propelled assets, repayment and on-line installment [3], [4]. Also, it might be connected into elective fields just as reasonable contracts [5], open administrations [6], web of Things (IoT) [7], name frameworks [8] and security administrations [9]. Those fields support blockchain in different manners by which. It's been demonstrated that diggers may come through bigger income than their legitimate offer through discourteous mining methodology [10]. Also, it's been demonstrated that protection break may furthermore occur in blockchain even clients exclusively make exchanges with their open key and individual key [11] Tschorsch et al. [12] made a specialized study viewing suburbanized computerized monetary forms just as Bitcoin. Nomura examination Institute made a specialized report with respect to blockchain [13].

The remainder of this paper is sorted out as pursues. Segment II presents blockchain plan. Area III shows run of the mill understanding calculations utilized in blockchain. Segment IV abridges the specialized difficulties and along these lines the ongoing advances during this space. Area V talks about some potential future bearings and segment VI finishes up the paper.

Blockchain Architecture

Fig 1 Architecture of Blockchain

Blockchain could be an arrangement of obstructs, that holds a whole rundown of managing records like standard open record [14]. Figure one represents partner degree case of a blockchain.

With a past square hash contained inside the square header, a square has only one parent square. Its cost taking note of that uncle squares (offspring of the square's progenitors) hashes would even be hang on in ethereum blockchain [15]. The essential square of a blockchain is named beginning square that has no parent square. We keep an eye on then legitimize the internals of blockchain in subtleties.

Blocks

A square comprises of the square header and the square body as appeared in Figure 2. Specifically, the square header incorporates:

(i) **Block rendition:** demonstrates that arrangement of square approval principles to pursue.

(ii) **Merkle tree root hash:**the hash worth of the considerable number of exchanges inside the square.

(iii) **Timestamp:** current time as seconds in Greenwich Mean Time since Jan one, 1970.

(iv) **nBits:** target limit of a real square hash.

(v) **Nonce:** partner degree 4-byte field, that occasionally begins with zero and will increment for each hash figuring (will be clarified in detail in Section III).

(vi) **Parent square hash:** A 256-piece hash worth that focuses to the past square.

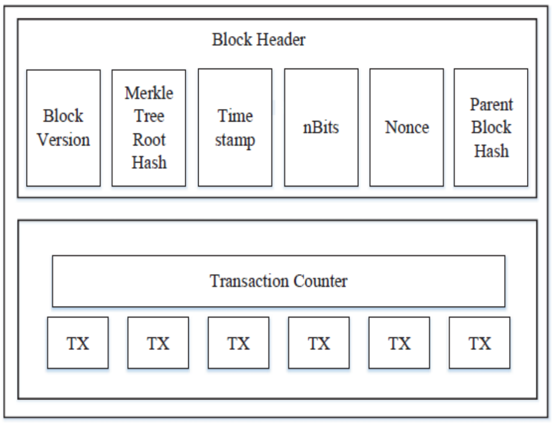


Fig 2 Block Architecture

Characteristics of Blockchain

In outline, blockchain has following key attributes:

•**Decentralization**. In standard brought together gathering activity frameworks, each gathering activity must be legitimate through the focal dependable organization (e.g., the national bank), unavoidably following to the worth and in this way the presentation bottlenecks at the focal servers. Differentiation to the concentrated mode, outsider isn't any more drawn out required in blockchain. Accord calculations in blockchain are acclimated keep up data consistency in disseminated arrange.

•**Persistency**. Exchanges are regularly substantial rapidly and invalid exchanges wouldn't be conceded by legitimate diggers. It's almost impractical to erase or rollback exchanges once they're encased inside the blockchain. Hinders that contain invalid exchanges likely could be found straightforwardly.

•**Anonymity.** Each client will act with the blockchain with a produced location, that doesn't uncover the $64000 personality of the client. Note that blockchain can't ensure the correct security safeguarding because of the characteristic requirement (subtleties will be referenced in area IV).

•**Auditability.** Bitcoin blockchain stores information with respect to client adjusts upheld the unexpended dealings Output (UTXO) model [2]: Any dealings must ask some past unexpended exchanges. When this managing is recorded into the blockchain, the condition of these alluded unexpended exchanges change from unexpended to spent. In this way, exchanges likely could be basically confirmed and followed.

CONSENSUS ALGORITHMS

Pow: (Proof of work) could be an agreement procedure utilized in the Bitcoin organize [2]. In PoW, each hub of the system is sagacious a hash worth of the square header. The square header contains a these days and excavators would change the these days regularly to incite totally extraordinary hash esteems. The agreement needs that the determined worth ought to be sufficient to or littler than a particular given worth.

PoS: (Proof of stake) is an imperativeness saving alternative rather than PoW. Diggers in PoS need to exhibit the obligation regarding proportion of cash. In particular, Blackcoin [16] uses randomization to foresee the following generator. It uses a condition that looks for the most insignificant hash a motivating force in mix with the range of the stake. Various blockchains grasp PoW at the begin and change to PoS a little bit at a time.

PBFT: (Practical byzantine fault tolerance) is a replication figuring to bear byzantine issues [17]. Hyperledger Fabric [18] utilizes the PBFT as its agreement computation since PBFT could manage up to 1/3 harmful byzantine generations.

DPOS: (Delegated proof of stake) is operator reasonable. Accomplices pick their specialists to deliver and endorse squares. thrown a poll out adequately. DPOS is the establishment of Bitshares [19].

Swell: Ripple [20] is an understanding count that utilizations overall trusted in subnetworks inside the greater framework. In the framework, center points are isolated into two sorts: server for taking an intrigue accord procedure and client for simply trading resources.

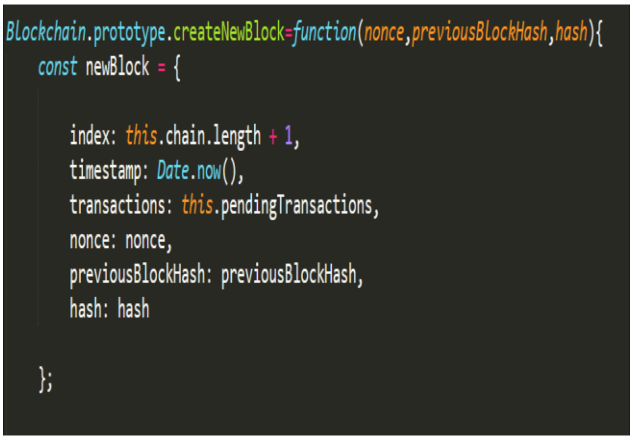
Practical Impementation of Blockchain : As An Instance Of Bitcoin Using Node.js

Block data:

Blockchain consists of immutable blocks that store data that can not be tampered with.Every block of this blockchain contains the following data:

1) **index:** this contains the index of the block in the block chain.

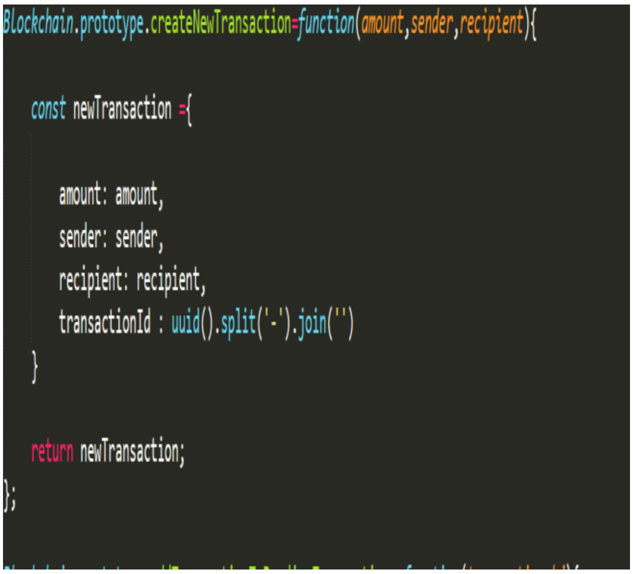
2) **timestamp**:this contains the time when block was created.
3) **transactions**:this contains all the transactions that got generated after the last block was created.
4) **nonce**:this value is used to get the desired hash of the block.
5) **previousBlockHash**: this contains the hash value of the previous block.This particular attribute of block makes it immutable.
6) **hash**:this contains the hash value of block.



```
Blockchain.prototype.createNewBlock=function(nonce,previousBlockHash,hash){
    const newBlock = {

        index: this.chain.length + 1,
        timestamp: Date.now(),
        transactions: this.pendingTransactions,
        nonce: nonce,
        previousBlockHash: previousBlockHash,
        hash: hash

    };
```

Fig 3 showing how to define new block function

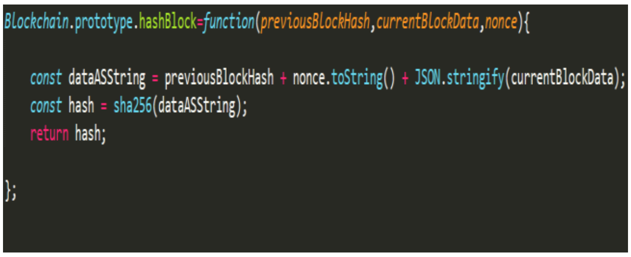
createNewTransaction:this function creates new transaction.New transaction has senderID,receiverID,transactionID and amount sent or received.



```
Blockchain.prototype.createNewTransaction=function(amount,sender,recipient){

   const newTransaction ={

       amount: amount,
       sender: sender,
       recipient: recipient,
       transactionId : uuid().split('-').join('')
   }

   return newTransaction;
};
```

Fig 4 showing how to create a new transaction function

Hash function: this function creates sha256 hash for each block. It takes previousBlockhash, currentBlockData(all the transactions of the block) and nonce as input and generates sha256 has string.

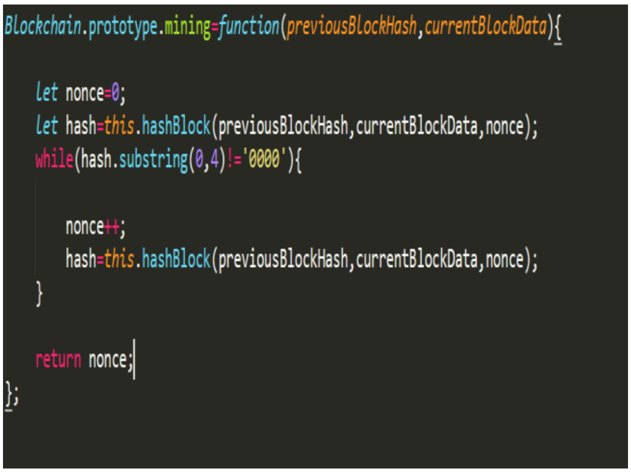


```
Blockchain.prototype.hashBlock=function(previousBlockHash,currentBlockData,nonce){

    const dataASString = previousBlockHash + nonce.toString() + JSON.stringify(currentBlockData);
    const hash = sha256(dataASString);
    return hash;

};
```

Fig 5 showing hash function

Mining: This function is bedrock for creation of blocks. New blocks will get created only when any of nodes in the network solves a puzzle. For example in this blockchain the puzzle is to obtain a hash value for the block so that it has four zeros in the starting of hash. For this the mining node takes previousBlockHash , all the pending transaction as new block's data as input and generates a value called nonce such that these three parameters(previousBlockHash ,currentBlockData and nonce) together gives a hash starting with four zeros and hence solves the puzzle.

This takes in a lot of computational power and hence longest chain in the blockchain is preferred as lots of computation has been done in creating all the blocks of longest chain. This is the very basis of consensus mechanism explained later.



```
Blockchain.prototype.mining=function(previousBlockHash,currentBlockData){

    let nonce=0;
    let hash=this.hashBlock(previousBlockHash,currentBlockData,nonce);
    while(hash.substring(0,4)!='0000'){

        nonce++;
        hash=this.hashBlock(previousBlockHash,currentBlockData,nonce);
    }

    return nonce;
};
```

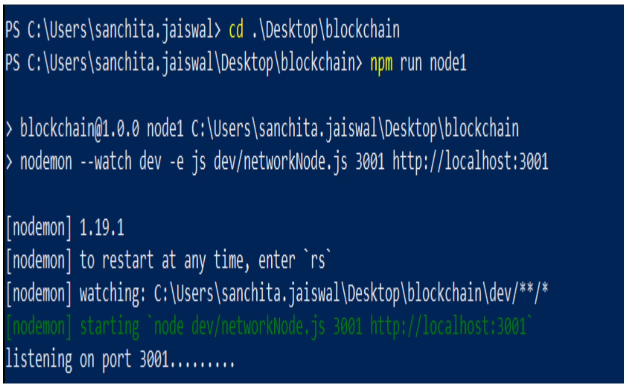
Fig 6 showing mining function

Node1 running: node js and following modules:

Nodemon

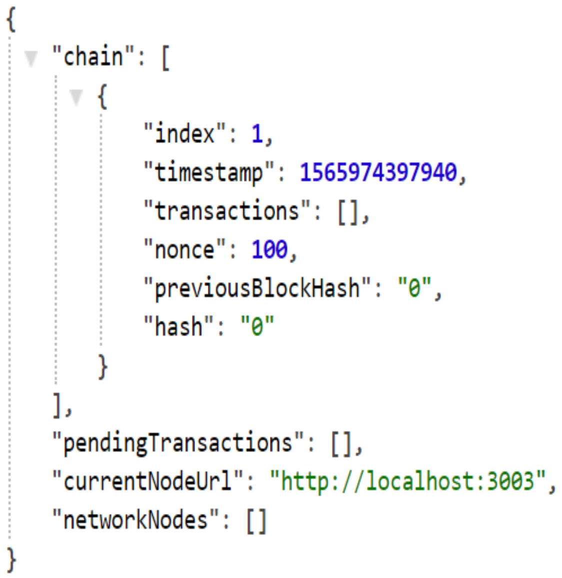
Body-parser

Request-promise



```
PS C:\Users\sanchita.jaiswal> cd .\Desktop\blockchain
PS C:\Users\sanchita.jaiswal\Desktop\blockchain> npm run node1

> blockchain@1.0.0 node1 C:\Users\sanchita.jaiswal\Desktop\blockchain
> nodemon --watch dev -e js dev/networkNode.js 3001 http://localhost:3001

[nodemon] 1.19.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: C:\Users\sanchita.jaiswal\Desktop\blockchain\dev/**/*
[nodemon] starting `node dev/networkNode.js 3001 http://localhost:3001`
listening on port 3001........
```

Fig 7 showing Blockchain with generic block:nonce=100,hash=0 and prevblockhash=0.



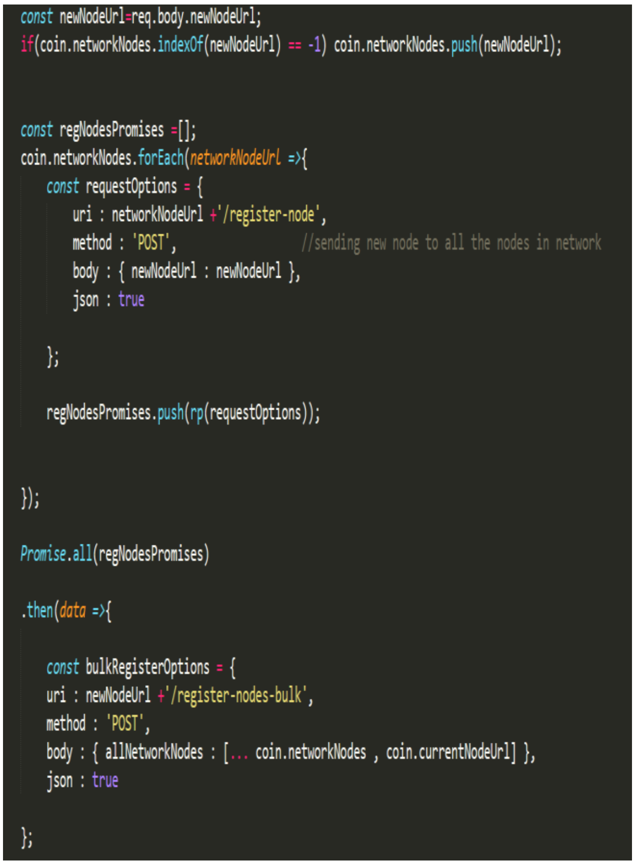
```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1565974397940,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    }
  ],
  "pendingTransactions": [],
  "currentNodeUrl": "http://localhost:3003",
  "networkNodes": []
}
```

Fig 8 showing Generic block of blockchain

New Node Registration and broadcast to the entire network(/register-and-broadcast-node):

To understand this api consider the scenario that node1, node2, node3 are nodes present in the blockchain (networkNodeUrl contains all the nodes present in the blockchain) and node4(newNodeUrl) wants to become a part of the blockchain. Then node4 can send request with its identity to any of the nodes in blockchain, let's say node1, node1 will register node4 and updates itself. Node1 then informs other nodes in the network about node4. All the nodes in blockchain(networkNodeUrl) now registers node4 to themselves by hitting their own api register-node api.

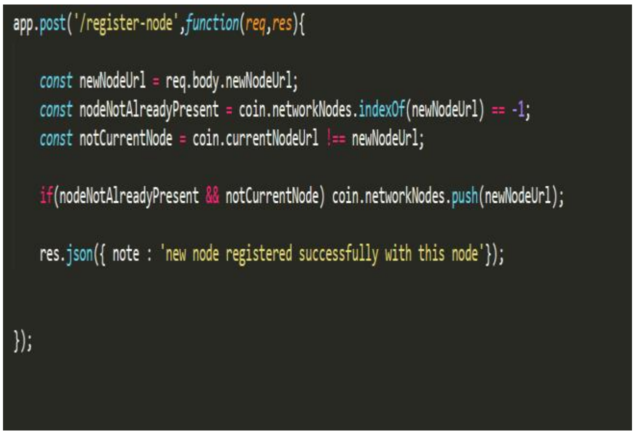
Subsequently node4 registers all the nodes in blockchain (networkNodeUrl) to itself by hitting its '/register-nodes-bulk'.



```
const newNodeUrl=req.body.newNodeUrl;
if(coin.networkNodes.indexOf(newNodeUrl) == -1) coin.networkNodes.push(newNodeUrl);


const regNodesPromises =[];
coin.networkNodes.forEach(networkNodeUrl =>{
    const requestOptions = {
        uri : networkNodeUrl +'/register-node',
        method : 'POST',             //sending new node to all the nodes in network
        body : { newNodeUrl : newNodeUrl },
        json : true

    };

    regNodesPromises.push(rp(requestOptions));


});

Promise.all(regNodesPromises)

.then(data =>{

    const bulkRegisterOptions = {
    uri : newNodeUrl +'/register-nodes-bulk',
    method : 'POST',
    body : { allNetworkNodes : [... coin.networkNodes , coin.currentNodeUrl] },
    json : true

};
```

Fig 9 showing registration and broadcasting of node to entire network

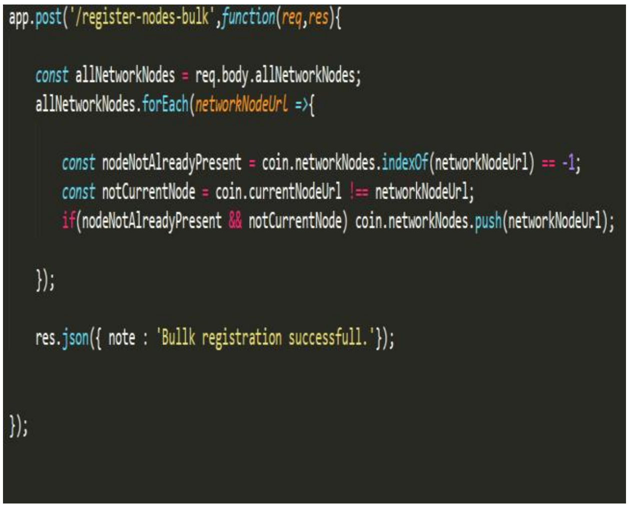
Register Node:



```
app.post('/register-node',function(req,res){

    const newNodeUrl = req.body.newNodeUrl;
    const nodeNotAlreadyPresent = coin.networkNodes.indexOf(newNodeUrl) == -1;
    const notCurrentNode = coin.currentNodeUrl !== newNodeUrl;

    if(nodeNotAlreadyPresent && notCurrentNode) coin.networkNodes.push(newNodeUrl);

    res.json({ note : 'new node registered successfully with this node'});


});
```

Fig 10 showing registration of node

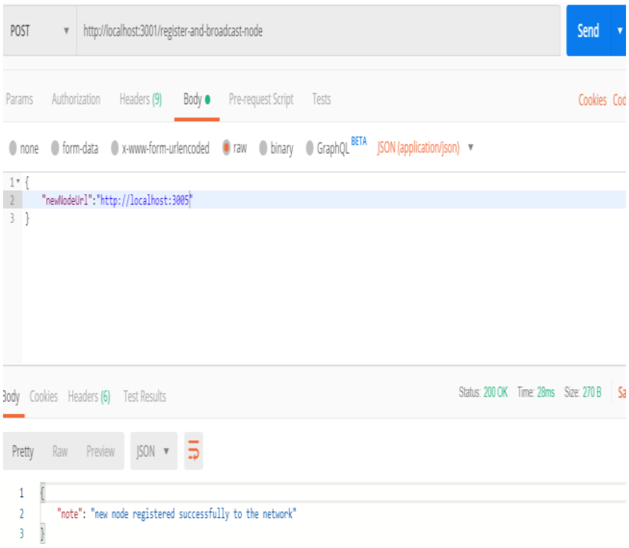
Bulk registration:



```
app.post('/register-nodes-bulk',function(req,res){

    const allNetworkNodes = req.body.allNetworkNodes;
    allNetworkNodes.forEach(networkNodeUrl =>{

        const nodeNotAlreadyPresent = coin.networkNodes.indexOf(networkNodeUrl) == -1;
        const notCurrentNode = coin.currentNodeUrl !== networkNodeUrl;
        if(nodeNotAlreadyPresent && notCurrentNode) coin.networkNodes.push(networkNodeUrl);

    });

    res.json({ note : 'Bullk registration successfull.'});


});
```

Fig 11 showing bulk registration of nodes

Demo of registration:



POST http://localhost:3001/register-and-broadcast-node

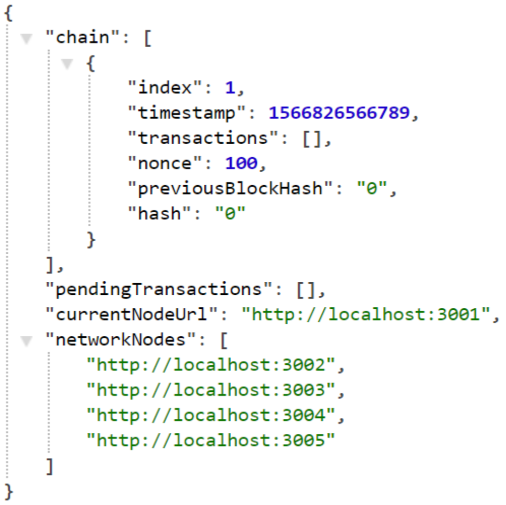
```
{
    "newNodeUrl":"http://localhost:3005"
}
```

Status: 200 OK Time: 28ms Size: 270 B

```
{
    "note": "new node registered successfully to the network"
}
```

Fig 12 is showing a demo of registration

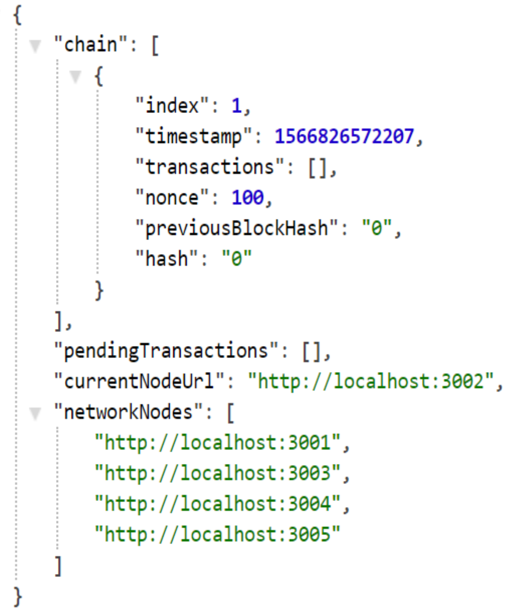
Node1:



```
{
  "chain": [
    {
        "index": 1,
        "timestamp": 1566826566789,
        "transactions": [],
        "nonce": 100,
        "previousBlockHash": "0",
        "hash": "0"
    }
  ],
  "pendingTransactions": [],
  "currentNodeUrl": "http://localhost:3001",
  "networkNodes": [
      "http://localhost:3002",
      "http://localhost:3003",
      "http://localhost:3004",
      "http://localhost:3005"
  ]
}
```

Fig 13 showing construction of node

Node2:

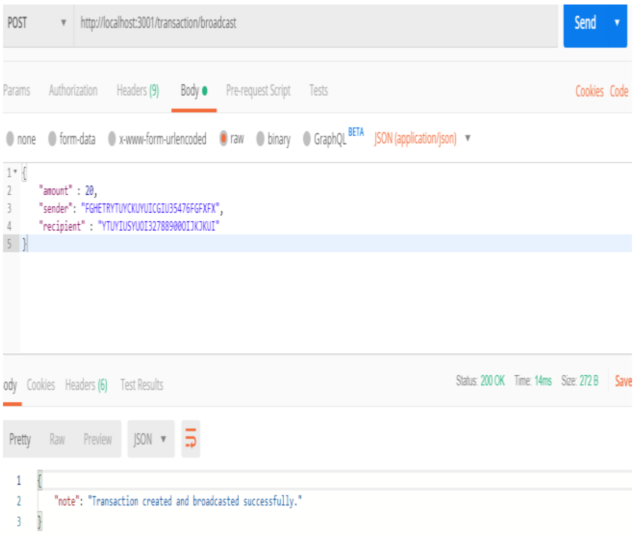


```
{
  "chain": [
    {
        "index": 1,
        "timestamp": 1566826572207,
        "transactions": [],
        "nonce": 100,
        "previousBlockHash": "0",
        "hash": "0"
    }
  ],
  "pendingTransactions": [],
  "currentNodeUrl": "http://localhost:3002",
  "networkNodes": [
      "http://localhost:3001",
      "http://localhost:3003",
      "http://localhost:3004",
      "http://localhost:3005"
  ]
}
```

Fig 14 showing construction of node 2

Similarly other three nodes also store the same type of data. At this stage we have all the nodes connected to each other and aware about every other node. That means anything/transaction made between node1 and node2 will be known to every node in the blockchain.

HOW TRANSACTION WORKS IN BLOCKCHIAN:



POST http://localhost:3001/transaction/broadcast

raw JSON (application/json)

```
{
    "amount" : 20,
    "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
    "recipient" : "YTUYIUSYUOI32788900OIJKJKUI"
}
```

Status: 200 OK Time: 14ms Size: 272 B

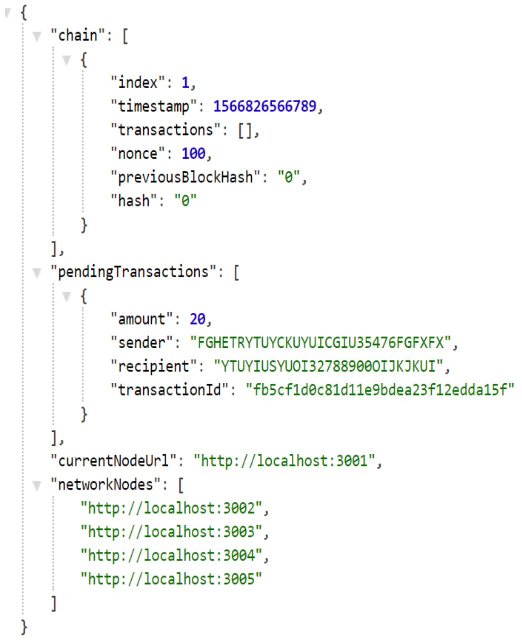
JSON

```
{
    "note": "Transaction created and broadcasted successfully."
}
```

Fig 15 showing working of transaction in blockchain

SenderID of node1 is “FGHETRYTUYCKUYUICGIU35476FGFXFX" and recipientID of node2 is "YTUYIUSYUOI32788900OIJKJKUI". Node1 sends 20 coins to node2.This transaction is generated and get reflected in every node’s pending transactions.

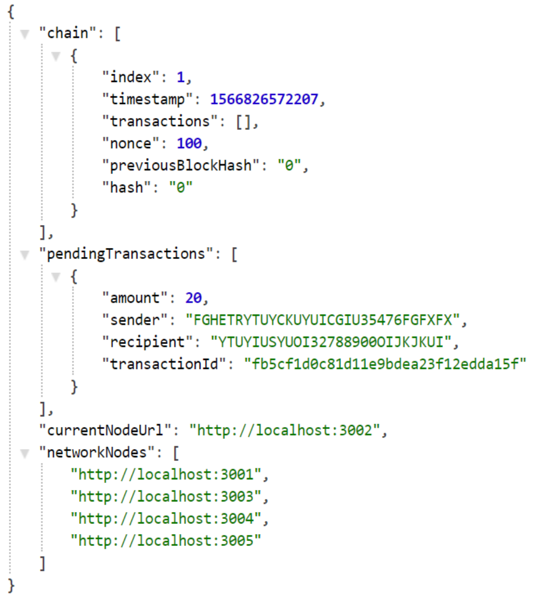
Node1 has following data in its blockchain:



```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1566826566789,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    }
  ],
  "pendingTransactions": [
    {
      "amount": 20,
      "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
    }
  ],
  "currentNodeUrl": "http://localhost:3001",
  "networkNodes": [
    "http://localhost:3002",
    "http://localhost:3003",
    "http://localhost:3004",
    "http://localhost:3005"
  ]
}
```

Fig 16 showing data inside node 1

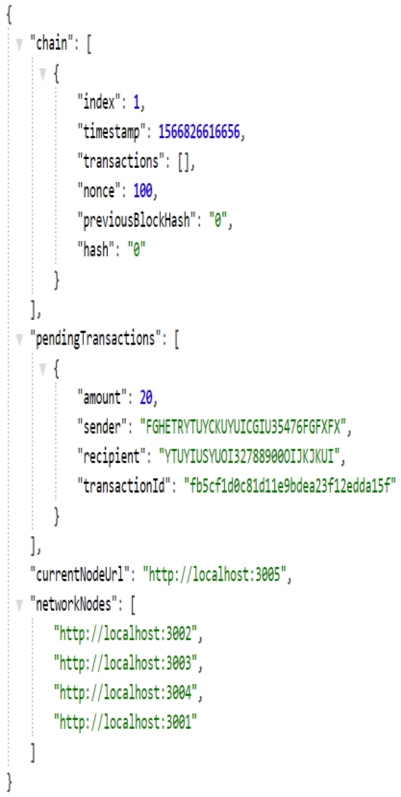
Node2 has following:



```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1566826572207,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    }
  ],
  "pendingTransactions": [
    {
      "amount": 20,
      "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
    }
  ],
  "currentNodeUrl": "http://localhost:3002",
  "networkNodes": [
    "http://localhost:3001",
    "http://localhost:3003",
    "http://localhost:3004",
    "http://localhost:3005"
  ]
}
```

Fig 17 showing inside data of node 2

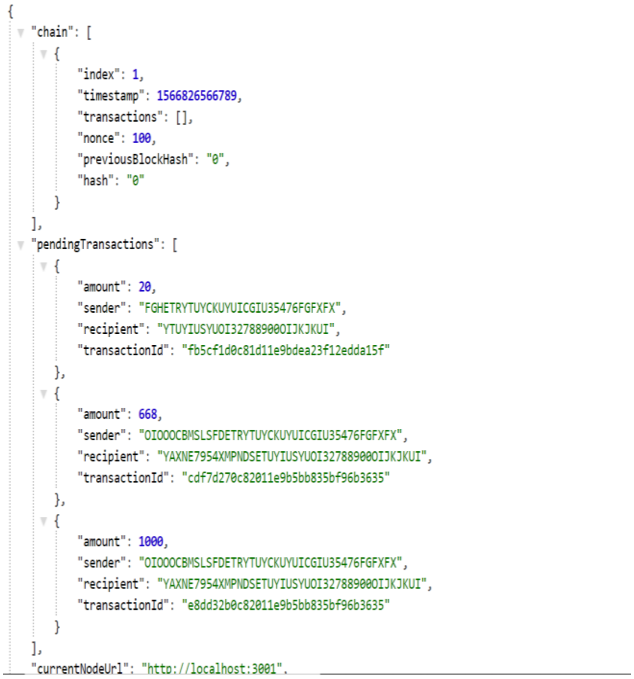
Node5 has the following data in its blockchain:



```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1566826616656,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    }
  ],
  "pendingTransactions": [
    {
      "amount": 20,
      "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
    }
  ],
  "currentNodeUrl": "http://localhost:3005",
  "networkNodes": [
    "http://localhost:3002",
    "http://localhost:3003",
    "http://localhost:3004",
    "http://localhost:3001"
  ]
}
```

Fig 18 showing inside data of node 5

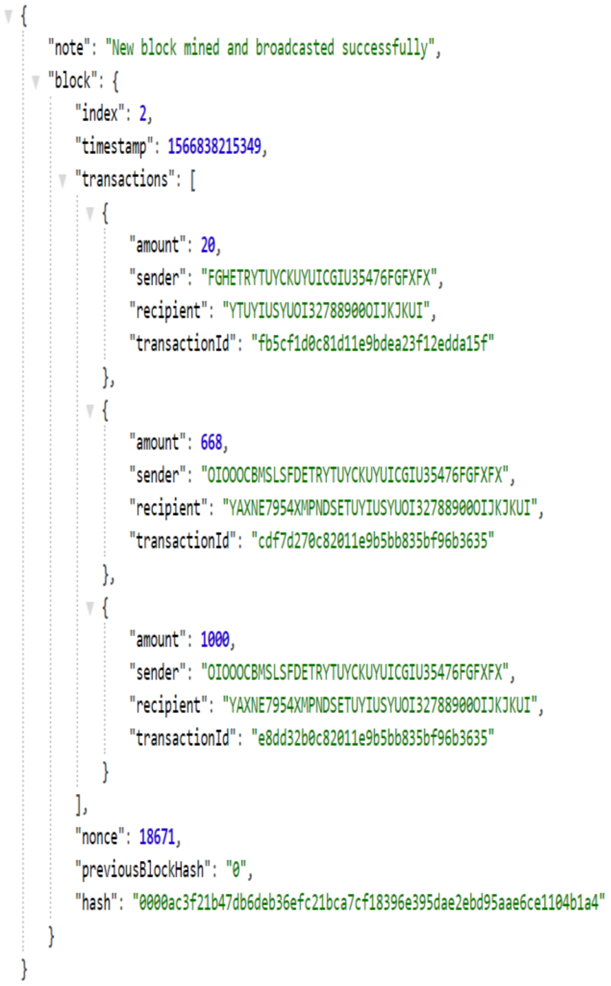
All (node1, node2, node5) nodes have same transactions, pending transactions and chain. Till now there is no block created. All transactions are currently present in pending transaction. Once any of the nodes perform mining a new block is created with index as 2 and transactions will contain all the pending transactions. Before demonstrating mining let us generate few more transactions (shown in following screenshots).



```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1566826566789,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    }
  ],
  "pendingTransactions": [
    {
      "amount": 20,
      "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
    },
    {
      "amount": 668,
      "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "cdf7d270c82011e9b5bb835bf96b3635"
    },
    {
      "amount": 1000,
      "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
      "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
      "transactionId": "e8dd32b0c82011e9b5bb835bf96b3635"
    }
  ],
  "currentNodeUrl": "http://localhost:3001",
```

Fig 19 showing different transactions

Only after mining a new block will get created and all these pending transactions will get in transaction object of the new block. Let us say node4 with ID “781d83f0c80611e9b5bb835bf96b3635” mines and generates second block as following.

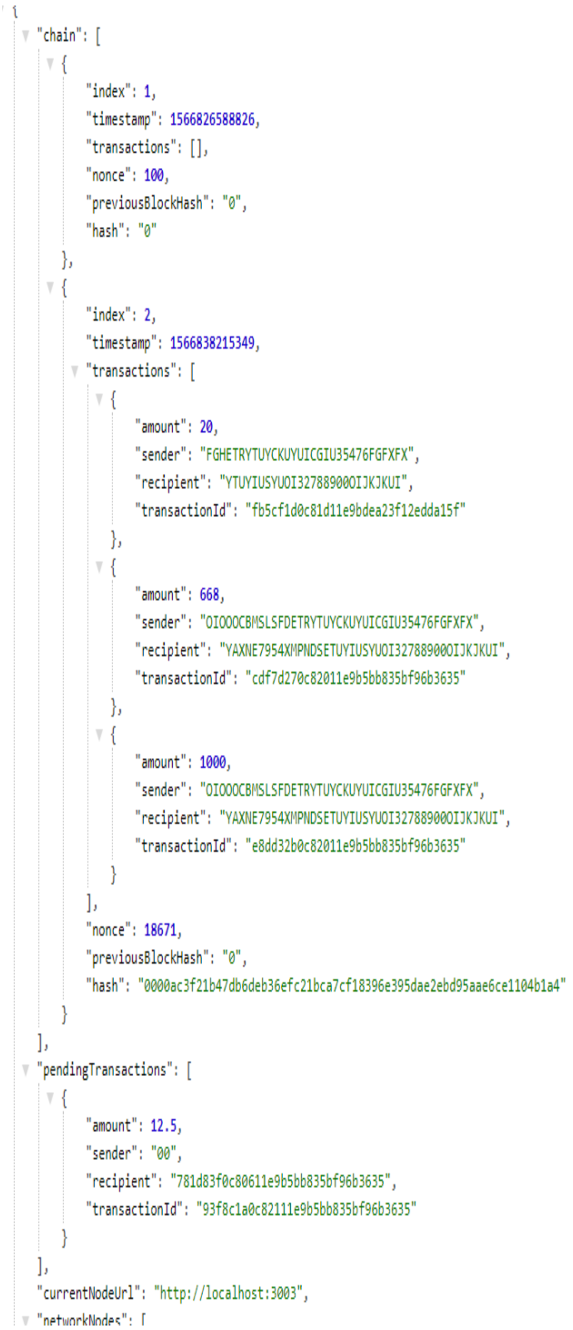


```
{
    "note": "New block mined and broadcasted successfully",
    "block": {
        "index": 2,
        "timestamp": 1566838215349,
        "transactions": [
            {
                "amount": 20,
                "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
                "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
                "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
            },
            {
                "amount": 668,
                "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
                "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
                "transactionId": "cdf7d270c82011e9b5bb835bf96b3635"
            },
            {
                "amount": 1000,
                "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
                "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
                "transactionId": "e8dd32b0c82011e9b5bb835bf96b3635"
            }
        ],
        "nonce": 18671,
        "previousBlockHash": "0",
        "hash": "0000ac3f21b47db6deb36efc21bca7cf18396e395dae2ebd95aae6ce1104b1a4"
    }
}
```

Fig 20 showing creation of block after mining function is executed

In bitcoin there is a mechanism to generate a reward of 12.5bitcoins to the miner. This transaction is generated after mining and goes into pending transaction.

New block after mining:



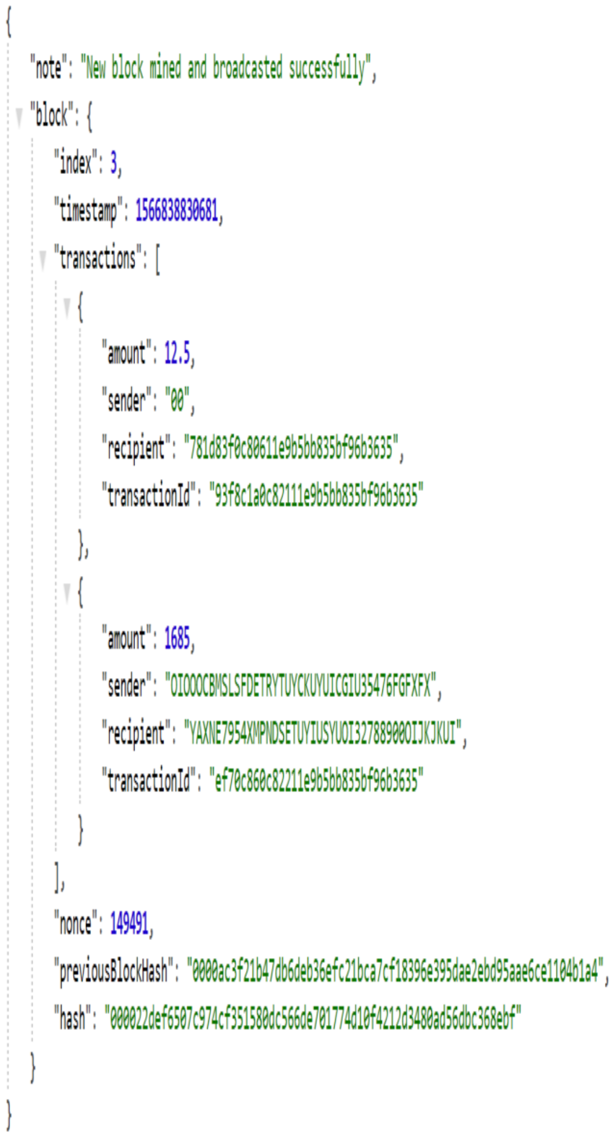
```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1566826588826,
      "transactions": [],
      "nonce": 100,
      "previousBlockHash": "0",
      "hash": "0"
    },
    {
      "index": 2,
      "timestamp": 1566838215349,
      "transactions": [
        {
          "amount": 20,
          "sender": "FGHETRYTUYCKUYUICGIU35476FGFXFX",
          "recipient": "YTUYIUSYUOI32788900OIJKJKUI",
          "transactionId": "fb5cf1d0c81d11e9bdea23f12edda15f"
        },
        {
          "amount": 668,
          "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
          "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
          "transactionId": "cdf7d270c82011e9b5bb835bf96b3635"
        },
        {
          "amount": 1000,
          "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
          "recipient": "YAXNE7954XMPNDSETUYIUSYUOI32788900OIJKJKUI",
          "transactionId": "e8dd32b0c82011e9b5bb835bf96b3635"
        }
      ],
      "nonce": 18671,
      "previousBlockHash": "0",
      "hash": "0000ac3f21b47db6deb36efc21bca7cf18396e395dae2ebd95aae6ce1104b1a4"
    }
  ],
  "pendingTransactions": [
    {
      "amount": 12.5,
      "sender": "00",
      "recipient": "781d83f0c80611e9b5bb835bf96b3635",
      "transactionId": "93f8c1a0c82111e9b5bb835bf96b3635"
    }
  ],
  "currentNodeUrl": "http://localhost:3003",
  "networkNodes": [
```

Fig 21 showing new block after mining

Here sender=00 implies it is reward transaction generated after mining.

Third block(check for hashes):When again mining happens it adds the reward transaction generated for node4 to third block and this process continues after every mining.

Reward Mechanism - As compensation for their efforts, **miners** are awarded **bitcoin** whenever they add a new block of transactions to the blockchain. The amount of new **bitcoin** released with each **mined** block is called the "block **reward**." The block **reward** is halved every 210,000 blocks, or roughly every 4 years

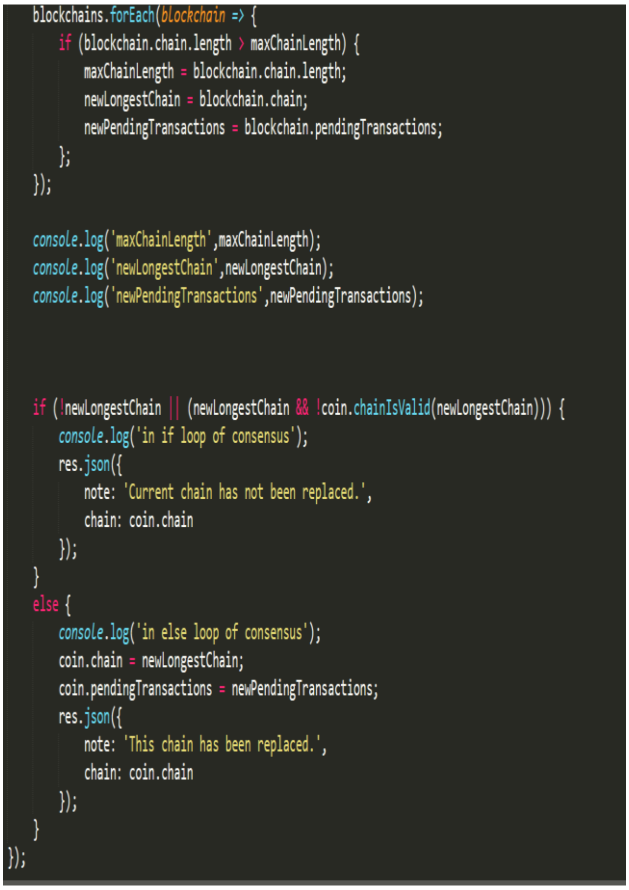


```
{
  "note": "New block mined and broadcasted successfully",
  "block": {
    "index": 3,
    "timestamp": 1566838830681,
    "transactions": [
      {
        "amount": 12.5,
        "sender": "00",
        "recipient": "781d83f0c80611e9b5bb835bf96b3635",
        "transactionId": "93f8c1a0c82111e9b5bb835bf96b3635"
      },
      {
        "amount": 1685,
        "sender": "OIOOOCBMSLSFDETRYTUYCKUYUICGIU35476FGFXFX",
        "recipient": "YAXNE7954XMPNDSETUYIUSYUOI327889000IJKJKUI",
        "transactionId": "ef70c860c82211e9b5bb835bf96b3635"
      }
    ],
    "nonce": 149491,
    "previousBlockHash": "0000ac3f21b47db6deb36efc21bca7cf18396e395dae2ebd95aae6ce1104b1a4",
    "hash": "000022def6507c974cf351580dc566de701774d10f4212d3480ad56dbc368ebf"
  }
}
```

Fig 22 showing rewarded transactions continued after ming

Consensus: This mechanism is very important for choosing the right blockchain from various nodes.

Consensus Mechanism - A **consensus mechanism** is a fault-tolerant **mechanism** that is used in computer and **blockchain** systems to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems, such as with cryptocurrencies.



```
    blockchains.forEach(blockchain => {
        if (blockchain.chain.length > maxChainLength) {
            maxChainLength = blockchain.chain.length;
            newLongestChain = blockchain.chain;
            newPendingTransactions = blockchain.pendingTransactions;
        };
    });

    console.log('maxChainLength',maxChainLength);
    console.log('newLongestChain',newLongestChain);
    console.log('newPendingTransactions',newPendingTransactions);



    if (!newLongestChain || (newLongestChain && !coin.chainIsValid(newLongestChain))) {
        console.log('in if loop of consensus');
        res.json({
            note: 'Current chain has not been replaced.',
            chain: coin.chain
        });
    }
    else {
        console.log('in else loop of consensus');
        coin.chain = newLongestChain;
        coin.pendingTransactions = newPendingTransactions;
        res.json({
            note: 'This chain has been replaced.',
            chain: coin.chain
        });
    }
});
```

Fig 23 showing consensus of longest chain in blockchain

Challenges

There is no uncertainty about the ability and limit of Blockchain, yet it suffers from some major issues. The extent of exchange is amplifying as time passes thus does the remaining task at hand of blockchain. Extreme exchanges make the framework massive. The size of squares is likewise an issue while conveying greater exchange. Being little in size, it can just contain a limited quantity of information at a time. That outcomes in deferral in little exchanges. In spite of the fact that the security is ensured in blockchain innovation anyway to be exact, the keys of open and private exchange can just protect a restricted measure of protection. While making the exchange, the namelessness of the clients is kept up. Be that as it may, it is appeared in [21], [5] that blockchain can't ensure the value-based protection since the estimations everything being equal and parities for every open key are freely obvious.

BLOCKCHAIN USES

The absolute first application and utilization of Blockchain was Bitcoin. In the present situation, money related circle has felt the nearness of blockchain innovation the most. Another application is "Keen Contracts". As the name recommends, a shrewd contract is a mechanized trade convention that executes the conditions of an understanding [22]. Making contracts and understandings savvy was however of long back, presently with the appearance of blockchain innovation, this can be figured it out. The reason of Finance being the biggest client of blockchain is the straightforwardness it gives to the gatherings of merchants and representatives while exchanging and making exchange. Exchanges occurring in any substance be it Private or Public can be put away in the squares and authenticity of the equivalent can be later checked. For doing endlessly the degenerate and malafied rehearses and understanding the fantasy of defilement free country can be acknowledged by bringing Blockchain in the standard and broadly utilizing it in every one of the areas eg. Decisions and Banking Sectors.

CONCLUSION

Blockchain is a progressive innovation which has changed the manner in which individuals cooperate with the Internet. In this digital world, where nothing is private, and no information is sheltered, Blockchain has indicated promising possibilities of being the best wagered of individuals managing the worth delicate items. In this paper we have talked about the nuts and bolts of blockchain innovation which can be utilized as a source of perspective for individuals new in this field. We have additionally laid out its conceivable application and the difficulties it faces. The point of this paper is to give a far reaching readymade thought of the working of blockchain innovation which can be utilized by understudies or whoever keen on getting acquainted with this progressive innovation.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online].Available: https://bitcoin.org/bitcoin.pdf

[2] "State of blockchain q1 2016: Blockchain funding overtakes bitcoin," 2016. [Online]. Available: http://www.coindesk.com/ state-of-blockchain-q1-2016/

[3] G. W. Peters, E. Panayi, and A. Chapelle, "Trends in crypto- currencies and blockchain technologies: A monetary theory and regulation perspective," 2015. [Online]. Available: http://dx.doi.org/10.2139/ssrn. 2646618

[1] G. Foroglou and A.-L. Tsilidou, "Further applications of the D. Lee Kuo Chuen, Ed., Handbook of Digital Currency, 1st ed. Elsevier, 2015. [Online]. Available: http://EconPapers.repec.org/RePEc: eee:monogr:9780128021170

[2] V. Buterin, "A next-generation smart contract and decentralized application platform," white paper, 2014.

[3] P. Vasin, "Blackcoins proof-of-stake protocol v2," 2014. [Online]. Available:https://blackcoin.co/blackcoin-pos- protocol-v2-whitepaper.pdf

[4] blockchain," 2015.

[5] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy- preserving smart contracts," in Proceedings of IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858.

[6] B. W. Akins, J. L. Chapman, and J. M. Gordon, "A whole new world: Income tax considerations of the bitcoin economy," 2013. [Online]. Available: https://ssrn.com/abstract=2394738

[7] Y. Zhang and J. Wen, "An iot electric business model based on the protocol of bitcoin," in *Proceedings of 18th International Conference on Intelligence in Next Generation Networks (ICIN)*, Paris, France, 2015, pp. 184–191.

[8] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in Proceedings of 11th European Conference on Technology Enhanced Learning (EC-TEL 2015), Lyon, France, 2015, pp. 490–496.

[9] C. Noyes, "Bitav: Fast anti-malware by distributed blockchain consensus and feedforward scanning," arXiv preprint arXiv:1601.01405, 2016.

[10] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg, 2014, pp. 436– 454.

[11] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2014, pp. 15–29.

[12] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 2084– 2123, 2016.

[13] NRI, "Survey on blockchain technologies and related services," Tech. Rep., 2015. [Online]. Available http://www.meti.go.jp/english/press/ 2016/pdf/0531 01f.pdf

[14] D. Lee Kuo Chuen, Ed., Handbook of Digital Currency, 1st ed. Elsevier, 2015. [Online].Available:

http://EconPapers.repec.org/RePEc: eee:monogr:9780128021170

[15] V. Buterin, "A next-generation smart contract and decentralized application platform," white paper, 2014.

[16] P. Vasin, "Blackcoins proof-of-stake protocol v2," 2014. [Online]. Available:https://blackcoin.co/blackcoin-pos- protocol-v2-whitepaper.pdf

[17] C. Miguel and L. Barbara, "Practical byzantine fault tolerance," in Proceedings of the Third Symposium on Operating Systems Design and Implementation, vol. 99, New Orleans, USA, 1999, pp. 173–186.

[18] "Hyperledger project," 2015. [Online]. Available: https://www.hyperledger.org/

[19] D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," Ripple Labs Inc White Paper, vol. 5, 2014.

[20] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," in Proceedings of the 2013 Conference on Internet Measurement Conference (IMC'13), New York, NY, USA, 2013.

[21] N. Szabo, "The idea of smart contracts," 1997.