Adoption of Blockchain to build a Cryptocurrency for Ledger Systems

Salil Abrol, Ajay Dureja, Aman Dureja

Abstract: ***The idea of the cryptocurrency was to decentralize the currency system by establishing transactions over distributed peer to peer network [1]. The technology of Blockchain was adopted to achieve this motive. The term blockchain comes from the idea of list of blocks, growing continuously over time wherein every block carries the data relating to the transactions and data regarding the cryptographical linkage using secure hash algorithms and the protocols [2]. Through this paper, we have shown the implementation of the blockchain technology so as to build the cryptocurrency. While building up the cryptocurrency, called the 'SantCoin', the idea about how this technology can revolutionize the traditional existing ledger systems can be upgraded so as to implement secure means of transactions over a distributed network. This implementation work suggests the use of technology in almost every governing body so that they can secure themselves and limit the dependency on human resource to do their central authoritative work [3].***

Keywords: Block, Blockchain, Cryptocurrency, Cryptocurrency linkage, Currency, Distributed network, Ledger, Peer to peer networks, Transactions, SantCoin, Server.

I. INTRODUCTION

Through the course of this paper, we will see how a cryptocurrency can be built up based on the technology of Blockchain [4]. By building up the cryptocurrency through this technology, it is depicted that in similar way the Blockchain can be employed to revolutionize the task in almost every aspect of every sector [5], whether it may be a banking sector [6], accounting sector, property sector, or any other business-oriented tasks. Through this technology used in this implementation, it is depicted that there is no central authority and everything is being carried out over distributed peer to peer network and transactions are getting stored in a continuously growing list of records called a Blockchain, and there it is secured by secure hash algorithm SHA256 and various rules are applied so as to prevent the blockchain being hacked. No peer is acting as central server, all have equal access to the chain and is maintained in every peer's local memory. Go through the implementation work, and you will certainly find it helpful for using this technology in any of your domain.

Revised Manuscript Received on October 15, 2019
***** Correspondence Author
Salil Abrol, Department of Computer Science and Engineering, PDM University, Bahadurgarh (Hr.), India. Email: salilabrol@hotmail.com
Ajay Dureja, Department of Computer Science and Engineering, PDM University, Bahadurgarh (Hr.), India. Email: ajaydureja@gmail.com
Aman Dureja, Department of Computer Science and Engineering, PDM University, Bahadurgarh (Hr.), India. Email: amandureja@gmail.com

The Paper is organized in 3 chapters. Chapter 2 discusses about the IDEs, and tool used. It also discusses about the approach that is to be followed to build the cryptocurrency. In this chapter the code is also given to implement the cryptocurrency. Chapter 3 provides the simulation results of the python code with screenshots of every domain of the code and finally Chapter 4 concludes the research paper by discussing about the future works that can be done based on the knowledge provided by this paper.

II. METHODOLOGY, IDEs, TOOLS AND IMPLEMENTATION

A. SPYDER IDE

Spyder is a great Integrated development environment which is itself developed in python for developing applications and constructs in python. It has a compiled environment and is a great tool for development and data exploration. It was initially released in October 2009. However stable release was found in 2019 only. The repository is available at github at github.com/spyder-ide/spyder [7].

B. Use of Spyder IDE And The Supporting Tools

Spyder IDE will be used to implement our blockchain in python. Along with the Spyder, we need to install two more tools i.e. the Flask, which is a framework that will provide tools and libraries such as Werkzueg and Jinja used to build a web application, and the other tool is Postman having a user-friendly UI used to deal with the HTTP Get and Post requests.The Flask can be installed by simply giving the command in the command prompt.

pip install Flask==0.12.2

while the Postman is downloaded from the official website of Postman i.e. www.getpostman.com.

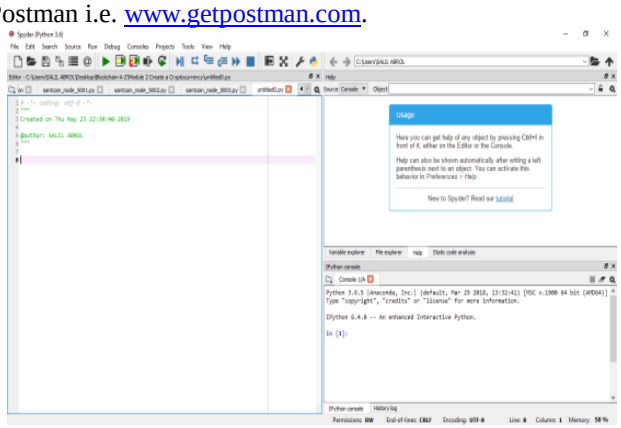

Fig. 1. Glimpse of Spyder IDE

On the lest side is the working environment to write the python code, and on the bottom right is the python console which shows the result of the simulation.
We proceed in three parts, Part 1 will show the building up of the Blockchain and Part 2 will show the mining of the blockchain and Part 3 will be for implementing the most important feature of blockchain that is the decentralization.

C. Building up the Blockchain

Before proceeding with the blockchain code implementation, we create two json files, transaction,json and nodes.json.

Contents of transaction.json are:

```
{
   "sender":"",
   "receiver":"",
   "amount":
}
```

Contents of nodes.json are:

```
{
   "nodes" : ["http://ip:port-number1",
            "http:// ip:port-number2",
            "http:// ip:port-number3"]
}
```

The nodes.json shows that there are 3 nodes in distributed p2p network, with port numbers port-number1, port-number2, port-number3.

So, we start the implementation by building up the blockchain. We build the blockchain in two parts, in the first part we just build the architecture of the blockchain and in part 2 we will make 2 functions to get the state of the blockchain in order to display it in the postman and other function to mine the blocks.

First, we will import some needed libraries like the datetime library because each block will have a time-stamp of when it was mined.

Second, we will use the hashlib library that will be used to hash the blocks.

Third, we will import the json library from which we will use he functions to encode the block.

Fourth, we will use the flask library's flask class which will be the pre made web application itself, and then we use the jsonify class to post the messages to interact with our blockchain.

After importing the required libraries, we build the basic architecture of the blockchain. We will define a class that will include all the methods and properties related to the blockchain. In this class, we will include the genesis block, then we will initialize a chain that will be done in the 'init' function, then we will make a create block function to add a new block, and which will be used to mine a block later on, and then we will add some tools to make sure that the blockchain is solid i.e. the chain which can not be hacked or broken. The initialization will be done by the 'init' function that we first create in the blockchain class and that function will take the 'self' as the parameter. The 'self' is the object of the class that is created when we will build the whole class. That means, 'self' is the instance of the class, and can be said as the blockchain itself, because our class defines the architecture of the blockchain, and thus its instance would be a fully functional blockchain. Inside the 'init' function, we will declare a list, and this list would be the chain containing the blocks, that would be initially empty when no block is mined, and within this 'init' method, we will make the genesis block by calling the function that will create a block and we call this function using the object 'self'. This function is defined in the class only, and takes two arguments, the first one is the proof of the block and the second argument is the previous hash i.e. the hash of the previous block. As we are making the genesis block, we keep the value of second argument as 0, because there is no previous hash for the genesis block. In the 'init' function, we make another list to store the transactions, that will take place.

The class will carry a bunch functions:

init' function to initialize the properties and create a genesis block.

- A function to create a block that will be used whenever we want to mine a block and this function will create a block having the properties such as index, timestamp, proof, previous hash and the transactions. Finally, this block will append the created block to the blockchain.
- A function to return the index of the previous block.
- A function to define the proof of work, and to implement the SHA256 algorithm. We can make any formula to encode the hashes of the blocks and convert them to string, and then encode them to hexadecimal code and store it in a separate variable.
- A function to encode the block that will be done using the 'dumps' function from the json class.
- A function to check if the chain is valid or not. The checking will be done on the basis of comparison between the value of the previous hash variable and the hash of the previous block which is computed. If they match, then chain is valid, else not.
- A function to add the transactions to the block. This will add the data to the list of transactions we created in the 'init' function and the transactions will carry the name of the sender, the name of the receiver, and the amount sender sent to the receiver.
- A function to add the nodes to the distributed peer to peer network on which the blockchain is working.
- A function to replace the chain, if there is a situation of competing chains on the network. The chain will be preferred which will have the longest length and obviously on the basis of the decision made by the consensus protocol.

D. Mining up the Blockchain

To mine the blockchain we choose the following steps:

- Create a web app using the flask.
- Create the address for the node on the port.
- Create blockchain class' instance.
- We mine a new block using the mine block function. Create a mine block function that mines the block, adds the transactions to the new block. The new block will be of-course created by calling the function that was created inside the class to create a new block.
- At last we check the validity of the blockchain.

E. Decentralizing the Blockchain

Decentralizing the blockchain only connects up all the nodes in a distributed peer to per network, deals with the collisions and chain replacements if needed and finally running the app. Now, the only thing to do is, we need to replicate the above python files by number times, there are the number of nodes in the network. Suppose, if there are 3 nodes in our example, then we will build 3 copies of these python files whch will be differentiated as per the port number on which they will run. To do this, we just need to add one small line at the end of each file:

# For Running the app
app.run(host = 'host-id', port = 'port-number1')
for every node, the port number will be different as:
app.run(host = 'host-id', port = 'port-number2')
app.run(host = 'host-id', port = 'port-number3')

III. SIMULATION RESULTS AND DISCUSSIONS

A. SIMULATION IN POSTMAN

We use Postman to simulate the blockchain on a distributed peer to peer network. The distributed peer to peer network will generally be established on two or more different nodes or computers, but to simulate them on the same machine, we have the POSTMAN simulator, where we can transfer GET and POST requests within the nodes and exchange the transactions [8].

To start with the simulation, first we run our python files on the console. In our example, there are 3 python files, namely santcoin_node_5001.py, santcoin_node_5002.py and santcoin_node_5003.py. The following figure shows the execution of santcoin_node_5001.py on Console1/A of SPYDER IDE.

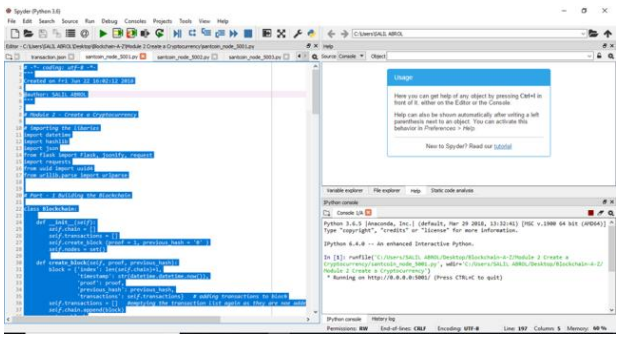

Fig. 2.Execution of santcoin_node_5001.py

Similarly, we can add the number of consoles depending on the number of nodes in our network and run the respective python files on the respective consoles. So, we run the other two python files also

```
In [1]: runfile('C:/Users/SALIL ABROL/Desktop/Blockchain-A-Z/Module 2 Create a
Cryptocurrency/santcoin_node_5002.py', wdir='C:/Users/SALIL ABROL/Desktop/Blockchain-A-Z/
Module 2 Create a Cryptocurrency')
 * Running on http://0.0.0.0:5002/ (Press CTRL+C to quit)
```

Fig. 3.Execution of santcoin_node_5002.py

```
In [1]: runfile('C:/Users/SALIL ABROL/Desktop/Blockchain-A-Z/Module 2 Create a
Cryptocurrency/santcoin_node_5002.py', wdir='C:/Users/SALIL ABROL/Desktop/Blockchain-A-Z/
Module 2 Create a Cryptocurrency')
 * Running on http://0.0.0.0:5002/ (Press CTRL+C to quit)
```

Fig. 4.Execution of santcoin_node_5003.py

Now, we open up the POSTMAN. The following figure shows the glimpse of POSTMAN, how it looks like.

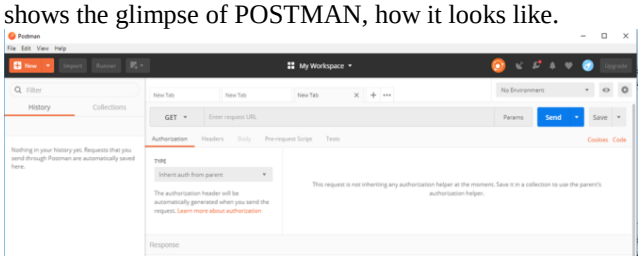

Fig. 5.Glimpse of POSTMAN

Now, by default there is always a Genesis Block present in the blockchain, which forms the first block of the blockchain. So, in the POSTMAN, there is a menu to select the type of request, we want to send. In this menu we will select the GET or POST request.

Also, there is a address bar, where we will put the URL of the node, to which we want to send the request. So, let us start by sending the GET request to the node 5001, that is our first node. We send the request by calling the method 'get_chain' we built in our python file. This method will display the current blockchain existing at that node.

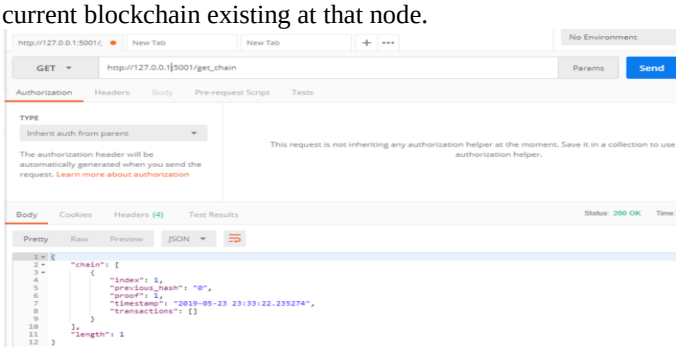

Fig. 6.get_chain request to node 5001

Here, it shows that there is one block in the blockchain which has the previous hash of 0, index is the block number, proof shows the proof of work, which is 1 for the genesis block, and transactions tab shows the transactions that are stored in the particular block. As in the genesis block, there are no transactions for the first time, so it is empty. Timestamp is another parameter that is stored in the block, that displays the time at which the block is mined.

Similarly, we call 'get_chain' method on all the nodes that is the nodes 5002 and 5003, we would get the genesis block in the blockchain.

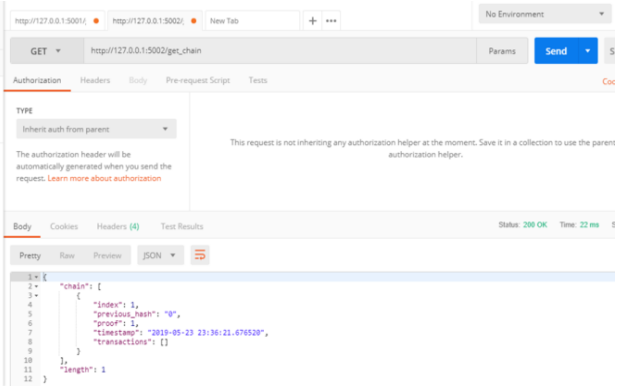

Fig. 7.get_chain request to node 5002

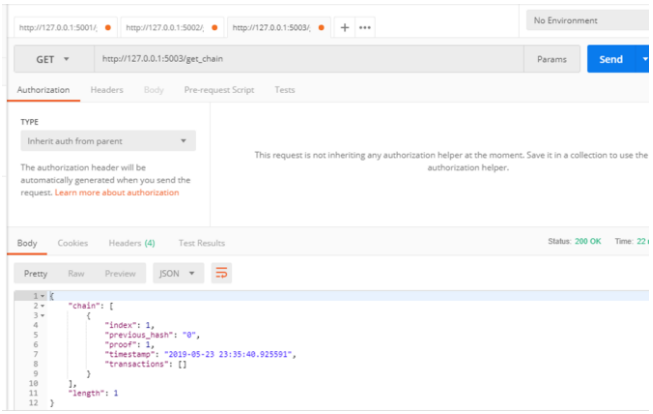

Fig. 8.get_chain request to node 5003

B. CONNECTING THE NODES TO FORM DISTRIBUTED P2P NETWORK

To connect the nodes, select the POST request from menu of the POSTMAN and call the method 'connect_node' that we built in the python file on each of the nodes that is here we call on nodes 5001, 5002 and 5003. So we connect node 5001 to 5002 and 5003, then we connect node 5002 to 5001 and 4003, and at last we connect node 5003 to 5001 and 5002. So now, we will have a fully connected network just like a fully connected graph.

We paste the contents of nodes.json file in the body section of the POSTMAN and run the requests to connect the nodes.

Fig. 9.Connecting node 5001 to 5002 and 5003

Fig. 10.Node 5001 connected to 5002 and 5003

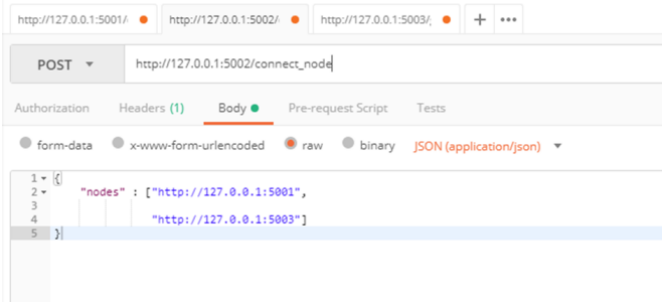

Fig. 11.Connecting node 5002 to 5001 and 5003

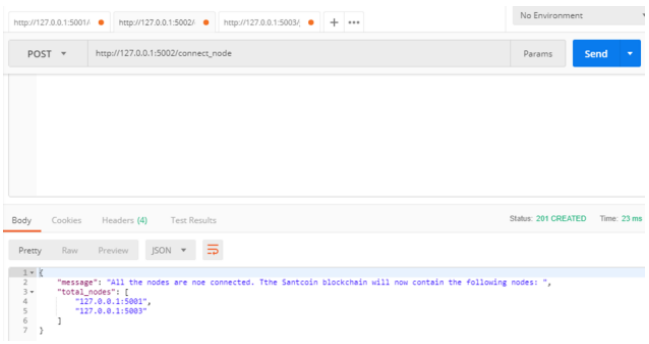

Fig. 12.Node 5002 connected to 5001 and 5003

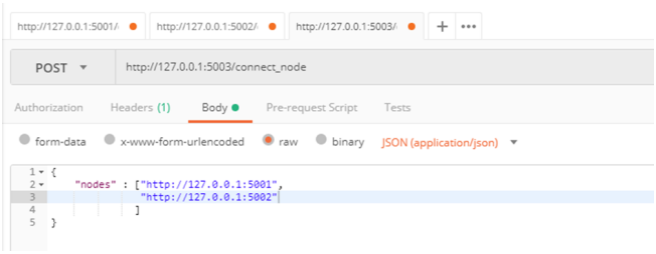

Fig. 13.Connecting node 5003 to 5001 and 5002

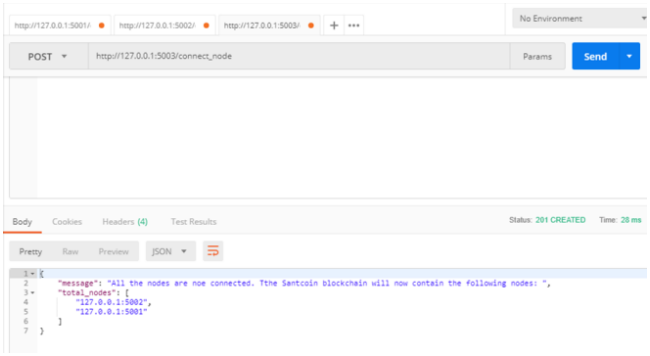

Fig. 14.Node 5003 connected to 5001 and 5002

By this time, our distributed peer to peer network is fully ready and now it is the time to start mining blocks, play transactions and load them onto the blocks and finally concatenating the mined blocks on the blockchain maintained in the network [9].

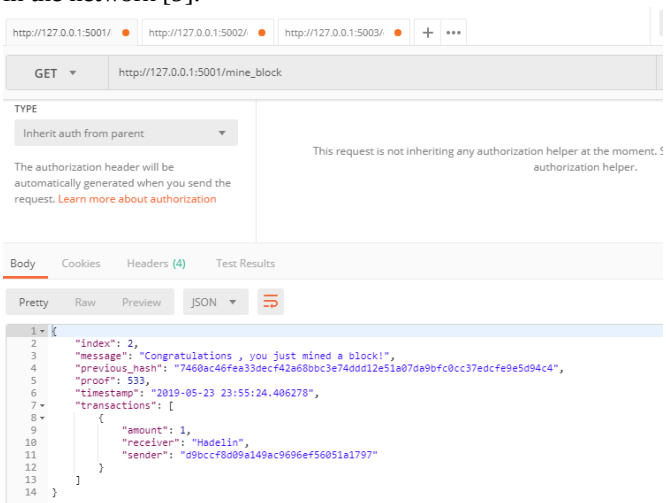

Fig. 15.Block mined by Node 5001

C. MINING A BLOCK

To mine a block, we need to switch to the GET request in the POSTMAN , and call the method 'mine_block'. The node on which the 'mine_block' will be called, will mine a block and will get a reward by the blockchain system to successfully mine a new block. So, to mine a block on the node 5001, switch to GET request and call the method 'mine_block' on this node.

The above figure shows that a block is mined successfully, and this block will be going to have index 2, that means it will be the second block in the blockchain as the first one was genesis block. Then it displays a message "congratulations, you just mined a block", it shows the hash of the previous block, proof of wor, timestamp, and contains one transaction, the transaction which will carry the reward for the node which mined the block, here the reward is for the node 5001, which is the receiver of the reward, and sender of the reward is the blockchain system, and the reward is 1 santcoin.

Now, we see the chain status by calling the method 'get_chain' on the node 5001.

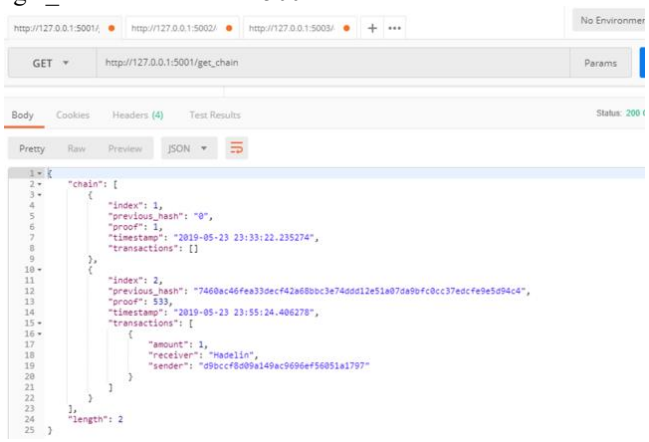

Fig. 16.Blockchain at Node 5001 has 2 blocks.

But if we see the chain status on other two nodes, it will show only one block, because we have not applied the consensus yet.

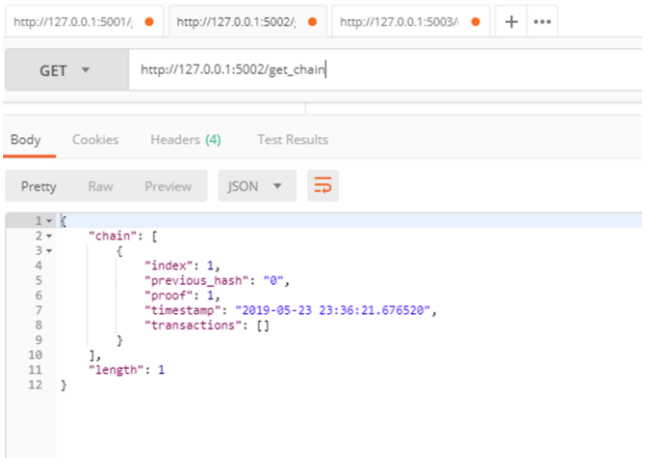

Fig. 17. Blockchain at node 5002 has 1 block.

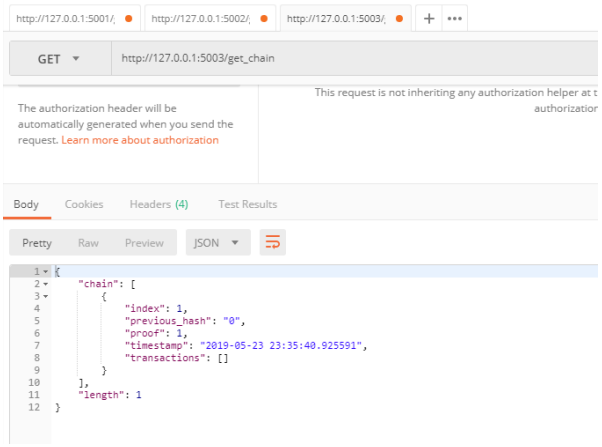

Fig. 18.Blockchain at node 5003 have 1 block.

To apply the consensus, we need to call 'replace_chain' method on the nodes 5002 and 5003 so that they also get the updated blockchain.

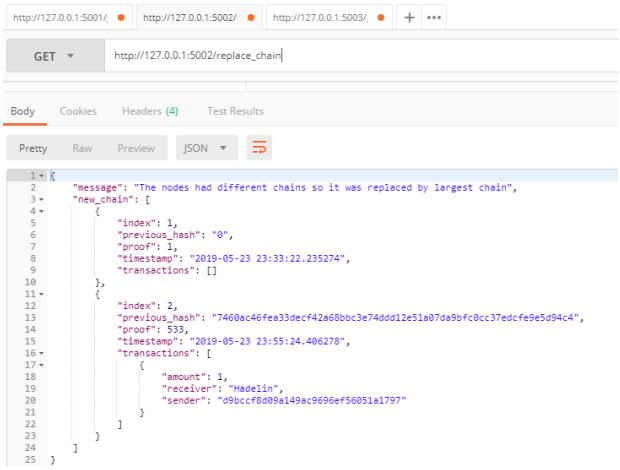

Fig. 19.Applying consensus at node 5002.

After, applying the 'replace_chain' method, the system shows the message, "The nodes had different chains so it was replaced by the largest chain", which is the fundamental phenomenon of the operation of consensus protocol. Now, if we call 'get_chain' method on node 5002, it will show the blockchain as it was on node 5001, that means now consensus is achived between the nodes 5001 and 5002.

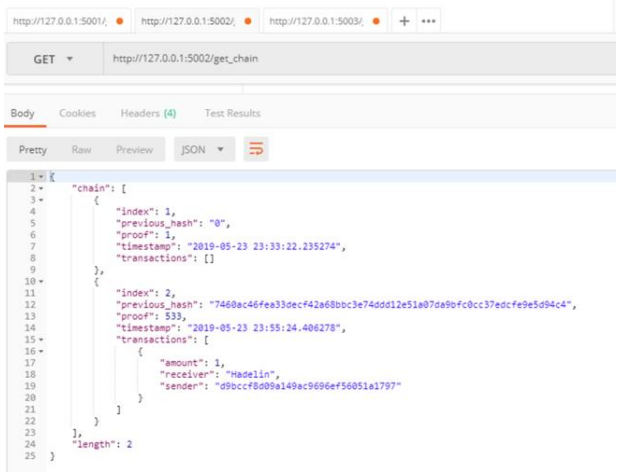

Fig. 20.Consensus achieved between node 5001 and 5002.

Similarly, we bring the consensus at node 5003, by calling ‘replace_chain’ and then we will see the blockchain with two blocks at node 5003 also.

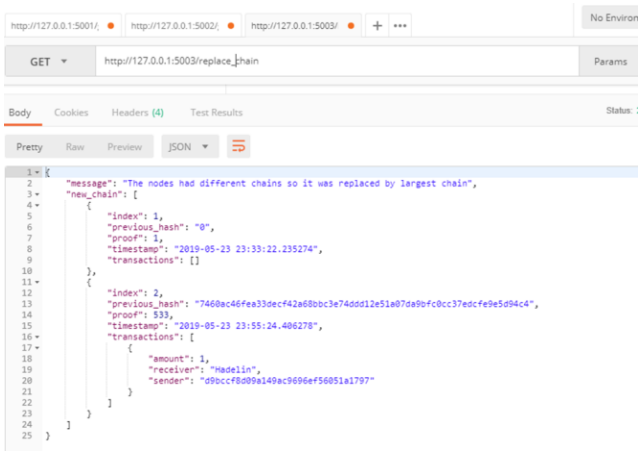

Fig. 21.Applying consensus at node 5003

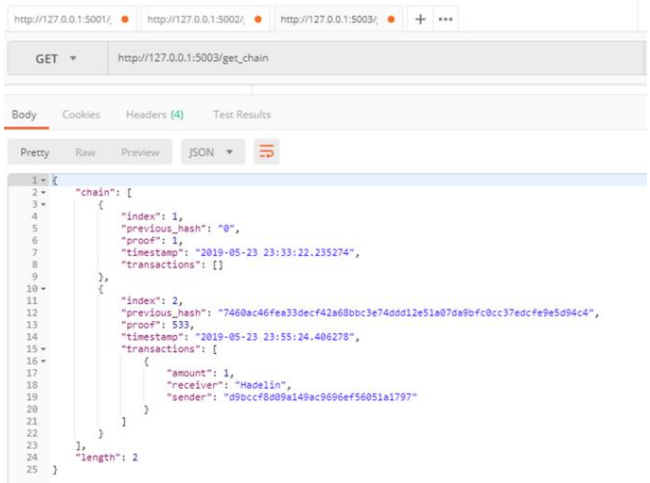

Fig. 22.Consensus achieved between node 5001,5002 and 5003.

Now, all the nodes in the distributed p2p network carry the same blockchain. So, now we can start transactions between the nodes[10]. These transactions will happen and get recorded on the peers itself, and whenever any new block is mined by any of the peers, these transactions and, also a new transaction of reward for mining the block for the peer which mined it, will be added to that new block and then consensus protocol will help to achieve consensus between all the nodes in the network ensuring that all of them have the same copy of the blockchain.

D. CARRYING OUT TRANSACTIONS BETWEEN THE NODES

To carry out the transactions between the nodes, we will use the transaction.json file, in which we built up the template for the transaction. It has a field ‘sender’, which will carry the public key of the sender peer, i.e. the peer which will send the santcoins. It has a field ‘receiver’, which will carry the public key of the peer which is going to receive the santcoinsand also it has the field ‘amount’ which will carry the number of santcoins that the sender will send to the receiver.

So, we copy and paste the transaction,json file’s contents in the body of the postman work area and simultaneously fill the fields sender, receiver and amount, and then send the POST request to the method ‘add_transaction’.

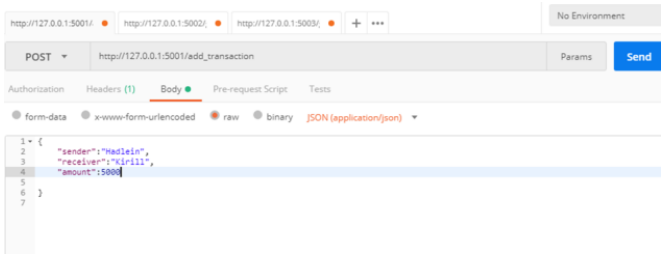

Fig. 23.Calling the add_transaction method on node 1.

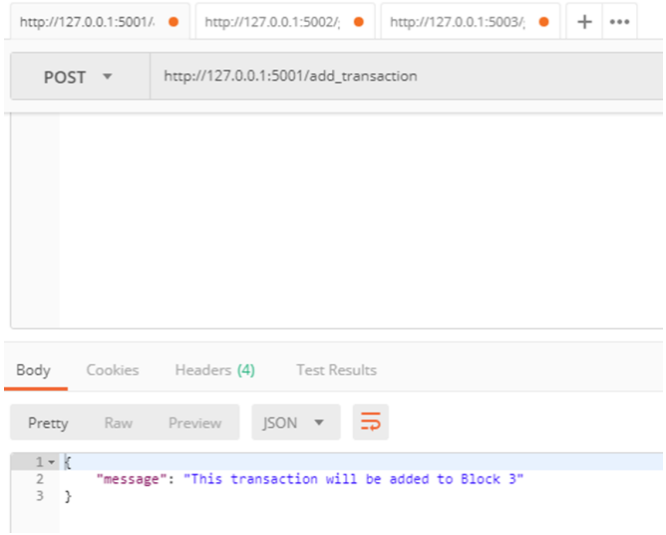

Fig. 24. On calling, transaction recorded.

On calling the ‘add_transaction’ method, the transaction is recorded at the peer 5001, and whenever a new block will be mined, the transaction will be added to the ‘transactions’ part of that block. Let us check it by mining a block by calling the ‘mine_block’ method at node 5001.

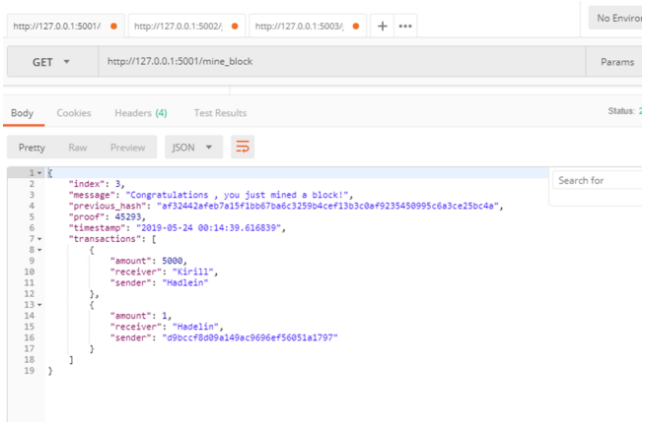

Fig. 25.Block mined by node 5001.

As, we can see, two transactions are added to this block 3, the first transaction is of 5000 santcoins that we sent from node 5001 to 5002 and the second transaction is the transaction of reward given by blockchain system to node 5001 for successfully mining the block. As there are 3 blocks now at node 5001, the consensus protocol will take care of achieving the consensus by calling ‘repplace_chain’ on the other nodes on the network so that they also carry the same blockchain as the node 5001 has.

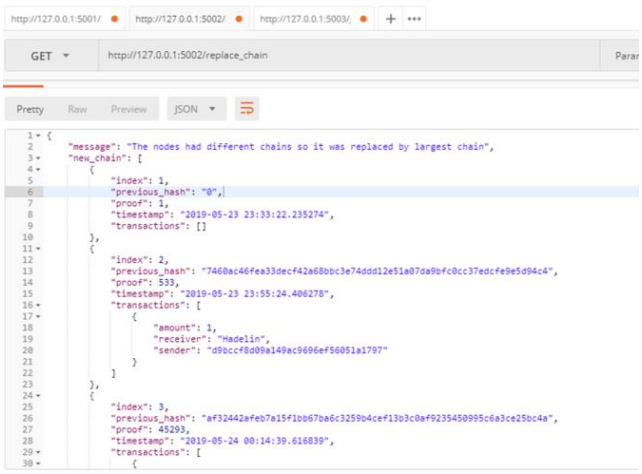

Fig. 26.Achieving consensus at node 5002.

Similarly, 'replace_chain' will help to achieve consensus at node 5003, and the blockchain at all the 3 nodes in the network will be as follows:


```
{
  "chain": [
    {
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2019-05-23 23:33:22.235274",
      "transactions": []
    },
    {
      "index": 2,
      "previous_hash":
"7460ac46fea33decf42a68bbc3e74ddd12e51a07da9bfc0cc3
7edcfe9e5d94c4",
      "proof": 533,
      "timestamp": "2019-05-23 23:55:24.406278",
      "transactions": [
        {
          "amount": 1,
          "receiver": "Hadelin",
          "sender":
"d9bccf8d09a149ac9696ef56051a1797"
        }
      ]
    },
    {
      "index": 3,
      "previous_hash":
"af32442afeb7a15f1bb67ba6c3259b4cef13b3c0af92354509
95c6a3ce25bc4a",
      "proof": 45293,
      "timestamp": "2019-05-24 00:14:39.616839",
      "transactions": [
        {
          "amount": 5000,
          "receiver": "Kirill",
          "sender": "Hadlein"
        },
        {
          "amount": 1,
          "receiver": "Hadelin",
          "sender":
"d9bccf8d09a149ac9696ef56051a1797"
        }
      ]
    }
  ],
  "length": 3
```


Now, let us add some ore transactions, first we send 10000 santcoins from node 5003 to 5002.

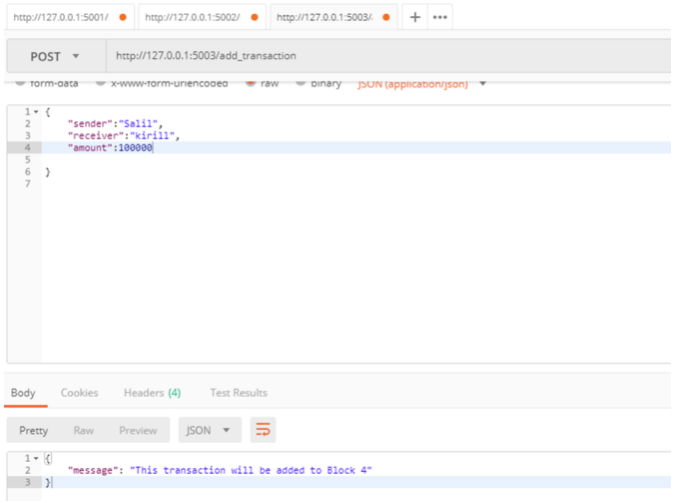

Fig. 27.Transaction will be added to block 4.

Now, we send 33 santcoins from Node 5002 to 5003. As after the previous transaction, no block is mined, so now there are two transactions waiting at the respective peers to be added to the block whenever a new block is mined.

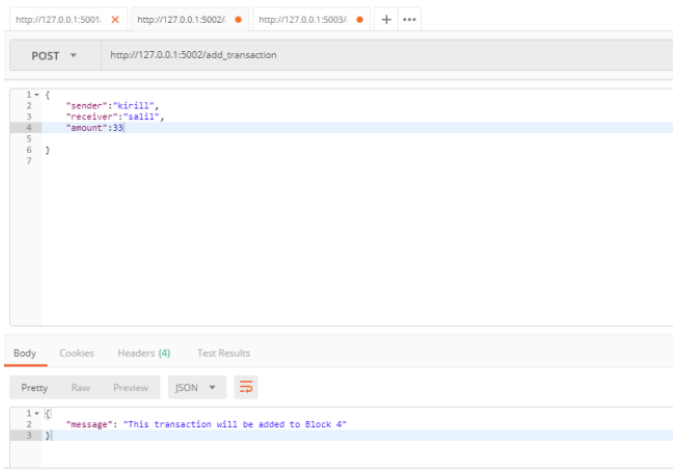

Fig. 28.Transaction will be added to block 4.

Now, let us mine a block by calling 'mine_block' on node 5003, so the waiting two transactions will be added to block 4 and one more transaction of reward by blockchain system to the node 5003 ill also be added. After this the node 5003 will have a blockchain with 4 blocks and other two nodes will have blockchain with 3 blocks, so the consensus protocol will come to play and call 'replace_chain' to update the largest chain on all the nodes of the network.

So, the chain will be:


```
{
  "chain": [
    {
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
```

```
        "timestamp": "2019-05-23 23:33:22.235274",
        "transactions": []
      },
      {
        "index": 2,
        "previous_hash":
"7460ac46fea33decf42a68bbc3e74ddd12e51a07da9bfc0cc3
7edcfe9e5d94c4",
        "proof": 533,
        "timestamp": "2019-05-23 23:55:24.406278",
        "transactions": [
          {
            "amount": 1,
            "receiver": "Hadelin",
            "sender":
"d9bccf8d09a149ac9696ef56051a1797"
          }
        ]
      },
      {
        "index": 3,
        "previous_hash":
"af32442afeb7a15f1bb67ba6c3259b4cef13b3c0af92354509
95c6a3ce25bc4a",
        "proof": 45293,
        "timestamp": "2019-05-24 00:14:39.616839",
        "transactions": [
          {
            "amount": 5000,
            "receiver": "Kirill",
            "sender": "Hadlein"
          },
          {
            "amount": 1,
            "receiver": "Hadelin",
            "sender":
"d9bccf8d09a149ac9696ef56051a1797"
          }
        ]
      },
      {
        "index": 4,
        "previous_hash":
"8713b05b5616c7f204976c41fb5d07f55561f58884b892151
d07cd7052751565",
        "proof": 21391,
        "timestamp": "2019-05-24 00:25:41.839079",
        "transactions": [
          {
            "amount": 100000,
            "receiver": "kirill",
            "sender": "Salil"
          },
          {
            "amount": 33,
            "receiver": "Salil",
            "sender": "kirill"
          },

          {
            "amount": 1,
            "receiver": "Salil",
            "sender":
"a721dc554ece4a87be453a3e92e96eee"
          }
        ]
      }
    ],
    "length": 4
}
```


IV. CONCLUSION AND FUTURE WORK

BLOCKCHAIN technology as we saw is a very intelligent method to solve the world's largest problem that is the data Security. The term blockchain was limited to original-ly for the cryptocurrencies but now it is a universal method that can be implemented to various projects and various sectors for implementing the security within the whole sector. Being an emerging technology, it will occupy our surroundings in every way and thereby providing us a secure digital world. This paper carried the understanding about the blockchain technology and concludes with show the practical implementation of this technology so as to build a new Cryptocurrency based on Blockchain, or can be adopted as a very efficient method to implement the Banking systems, ledger systems, accounting systems, payment wallets, credit and debit management in a totally decentralized manner without any central governing body.

REFERENCES

1. S. Haber, W.S. Stornetta "How to timestamp a digital document", Journal of cryptology, Vol.3, no. 2, pp.99-111, 1991.
2. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available:https://bitcoin.org/bitcoin.pdf
3. L. Lamport, R. Shostak, and M. Pease, "The byzantine generals prob-lem," ACM Transactions on Programming Languages and Systems(TOPLAS), vol. 4, no. 3, pp. 382–401, 1982.
4. 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.
5. DZ. Zheng, S. Xie, H. Die, X. Chen, H. Wang, "*An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*", In the Proceedings of the 2017 IEEE 6th International Conference on Big Data, Honolulu, HI, USA.
6. Kelly, Jemima (28 September 2016). "Banks adopting blockchain 'dramatically faster' than expected: IBM". Reuters. Archived from the original on 28 September 2016. Retrieved 28 September 2016.
7. Bheemaiah, Kariappa (January 2015). "Block Chain 2.0: The Renaissance of Money". *Wired*. Archived from the original on 14 November 2016. Retrieved 13 November 2016.
8. DeRose, Chris (26 June 2015). "Why the Bitcoin Blockchain Beats Out Competitors". *American Banker*. Archived from the original on 30 March 2016. Retrieved 18 June 2016.
9. Johnson DB, Maltz DA, Hu Y. The dynamic source routing protocol for mobile ad hoc networks (DSR).http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 2004. IETF Internet Draft .
10. Nian, Lam Pak; Chuen, David LEE Kuo (2015). "A Light Touch of Regulation for Virtual Currencies". In Chuen, David LEE Kuo (ed.). Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data. Academic Press. p. 319. ISBN 9 78-0-12-802351-8.J. Chen and Y. Kuo, 2009. Multipath routing protocol for network lifetime maximization in ad-hoc networks. Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, (WCNMC' 09), IEEE Xplore Press, Piscataway, NJ, USA, pp: 2713-2716. DOI:10.1109/ WICOM. 2009. 530 5828.

AUTHORS PROFILE

Salil Abrol pursed Bachelor of technology from PDM College of Engineering, MDU University in year 2017. He pursued his Master of Technology from PDM University in year 2019. He is currently working as Assistant Professor in PDM University**.**

Ajay Dureja pursuing Ph.D. from DCRUST, Murthal, Sonepat. He pursed Master of Technology from PDM College of Engineering, MDU University in year 2010. He pursed Bachelor of Technology from Bhiwani Institute of Technology & Sciences, MDU in year 2007. He is currently working as Assistant Professor in Department of Computer Science & Engineering, PDM University since 2010. He has published more than 20 research papers in reputed international journals including Scopus Imdexed and conferences including IEEE and it's also available online. His main research work focused on Internet of Vehicles, MANET and Image Processing. He has 9 years of teaching experience and 4 years of Research Experience.

Aman Dureja pursuing Ph.D. from GGSIPU, Dwarka, New Delhi. He pursed Master of Technology from PDM College of Engineering, MDU University in year 2010. He pursed Bachelor of Technology from Bhiwani Institute of Technology & Sciences, MDU in year 2007. He is currently working as Assistant Professor in Department of Computer Science & Engineering, PDM University since 2010. He has published more than 20 research papers in reputed international journals including Scopus Indexed and conferences including IEEE and it's also available online. His main research work focused on Machine Learning & Deep Learning . He has 9 years of teaching experience and 5 years of Research Experience.