Université Libre Bruxelles - Vrije Universiteit Brussel

Project Report

Communication Networks: Protocols and Architecture Project Block Chain

Authors :

Zhang Zheng
Zhang Boya
Zhai Peizhe
Wang Zhaoxiang

2018-2019

Contents

1 Introduction

Assume that we live in a town called the bit town, the ways of trading here has developed from barter, physical currency, symbol currency to central system virtual currency. The account book is managed by the mayor. Every resident uses virtual currency to pay. And the mayor is in charge of the maintenance of the account book. The problem with this modern financial system is: First, this system relies entirely on the personal credit of the mayor. If the mayor does not follow the rules and arbitrarily falsifies the account book, the entire monetary system will collapse. Second, if the mayor's home is on fire or the account book is stolen, it will also bring devastating effects to the financial system of the town. So one smart resident called Satoshi created a virtual currency system called bitcoin that does not depend on any central center, which will solve the problem above.

Bitcoin is a decentralized digital currency that enables instant payments to anyone, anywhere in the world. Bitcoin uses peer-to-peer technology to operate with no central authority: transaction management and money issuance are carried out collectively by the network. The biggest advantage of bitcoin is that it solved the root problem with conventional currency: the high cost of trust. Through the use of cryptographic proof, decentralized networks and open source software, bitcoin minimized the trust costs.[1]

2 Our tasks

Our basic requirement of the project is to create a blockchain network with a given topology. Further goal is to implement the mining algorithm in the verification.

2.1 Time schedule

- 25/11-1/12: Search and read articles on the internet, manage to understand the principles and mechanisms of bitcoin.
- 1/12-7/12: Write the code of networking and blockchain and debug it.
- 7/12-14/12: Debug the code, write the readme.md file and write the code comment.
- 14/12-21/12: Adjust and modify and write the report.

2.2 Achievement

We divided our project into two main parts: networking and blockchain. In the networking we implemented the network topology, realized communications between users and users as well as communications between users and the center. In the blockchain we made the algorithm work. Blocks are generated and linked together to become the blockchain. Every transaction information is recorded by every node. We managed to apply the mining to our system and increased its security level.

We will first talk about the networking part because the blockchain is based on it.

3 Networking

3.1 Overview

Networking is the practice of transporting and exchanging data between nodes over a shared medium in an information system. In our case, the networking contains multiple nodes(clients) and an authentication center. The authentication center will authenticate the clients. When a client is authenticated, the user is able to create a transaction through blockchain.

3.2 Network topology

In the networking part, we first set up the given network topology as the figure1. The network topology contains one center and six clients, and each client has a different router. Every node (clients and authentication server) will read from configuration file its IP address. We created a host.ini file for every node, where we will define the neighbors of every client.

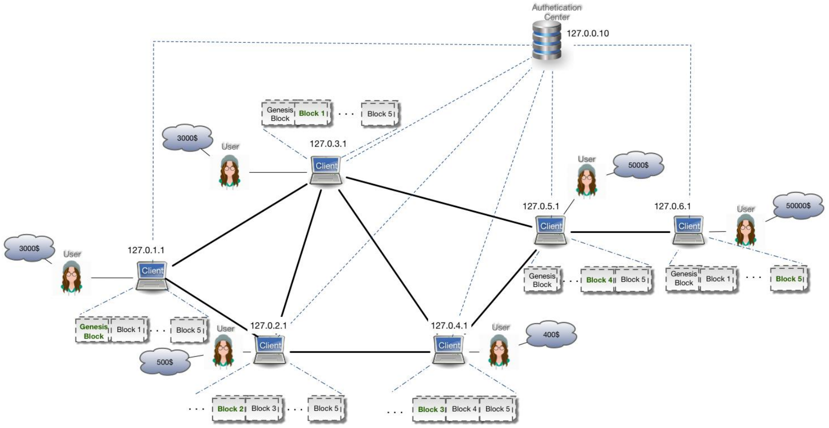


Figure 1: Network Topology

For each node, there is a host.ini containing the information of its IP address and neighbours. Thus, we can implement the network topology. We take host1.ini for client1 and host2.ini for client2 as examples.

1. Client1 (host1.ini)

```
[node]
ip_address=127.0.1.1
port=5001

[center]
ip_address=127.0.0.10
port=5001

[neighbours]
127.0.2.1,5001
127.0.3.1,5001
```

2. Client2 (host2.ini)

```
[node]
ip_address=127.0.2.1
port=5001

[center]
ip_address=127.0.0.10
port=5001

[neighbours]
127.0.1.1,5001
127.0.3.1,5001
127.0.4.1,5001
```

3.3 Registration Phase

Our second task is the registration phase. We give the authentication center a list of all clients of the network and a shared secret (username and password of clients). The list is stored in file center.ini.

The client will first request a nonce from the authentication center. The center will check the username and reply a nonce to the client. Then the client will send the username with the hash of the nonce plus the password to the center. The center will also perform the hash of the nonce and the password, then verify if two number are the same. If the same, the client can login. After the client has signed in, the authentication center will record the IP address of the node. The sequence diagram of communication between the authentication center and clients is shown in figure 2

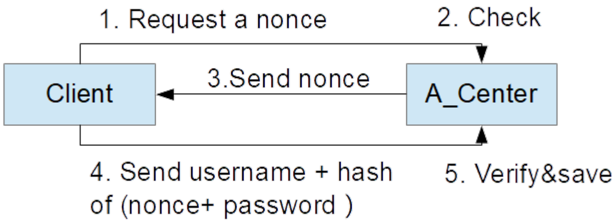


Figure 2: Registration Phase

3.4 User Interface

After logging in, the user will see the user interface options, where he or she can choose what to do. We included these options:

- Account information, where the user will see his or her balance
- Make transfer, where the user can make transfer to others
- Display the blockchain, where the user can see the current blockchain
- Update the blockchain, where the user can request a current blockchain from neighbours if he or she has logged off for some time
- Log-off, where the user can sign out

3.5 Broadcast among nodes

The message transfer between clients is done through broadcasting. The main idea is to broadcast the information from one node to its neighbours, and the next node will also perform the same procedure. Thus the information is broadcast to every node. The block can be broadcast among the clients online. Also, for clients who has been offline for some time, he or she can also request for updating information from other clients.

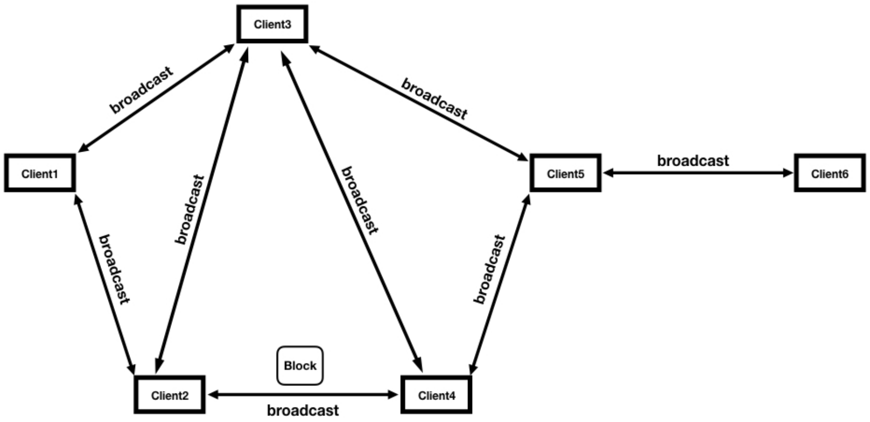


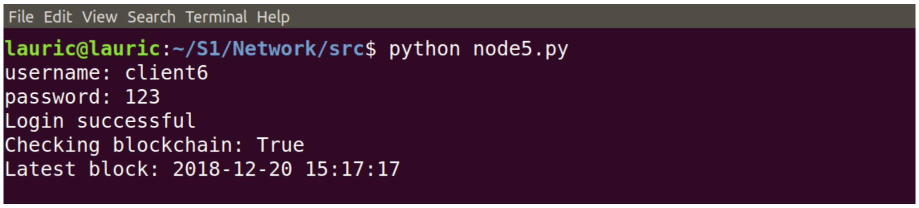
Figure 3: Broadcast

3.6 Result

3.6.1 Registration

For the registration, if the client inputs the right username and password, the center will recognize the user has logged in and will show the user interface. If not, the user can try again or exit.

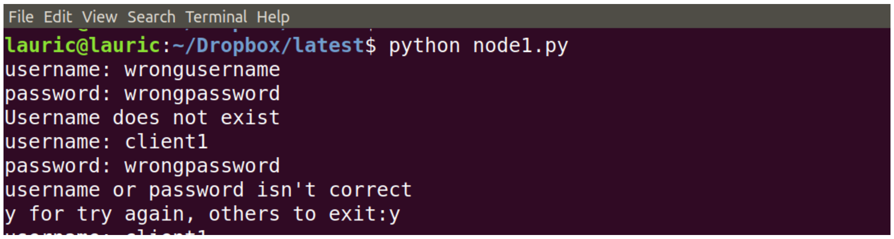
- Good username and password



```
lauric@lauric:~/S1/Network/src$ python node5.py
username: client6
password: 123
Login successful
Checking blockchain: True
Latest block: 2018-12-20 15:17:17
```

Figure 4: Registration Phase 1

- Wrong username and password

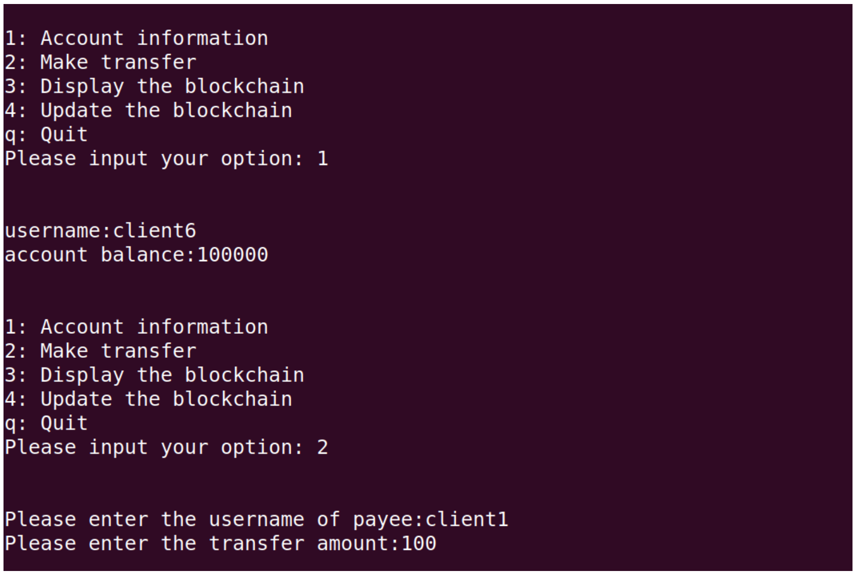


```
lauric@lauric:~/Dropbox/latest$ python node1.py
username: wrongusername
password: wrongpassword
Username does not exist
username: client1
password: wrongpassword
username or password isn't correct
y for try again, others to exit:y
```

Figure 5: Registration Phase 2

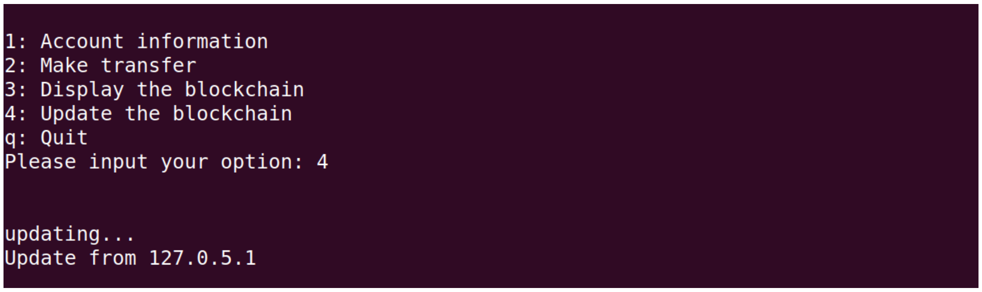
3.6.2 User Interface

These are just examples of the user interface in figure 6.We can see that the user checked the account information and then made a transfer option. We can also see the result of a client just logged in ask for the updated blockchain from neighbours from figure 7.



```
1: Account information
2: Make transfer
3: Display the blockchain
4: Update the blockchain
q: Quit
Please input your option: 1


username:client6
account balance:100000


1: Account information
2: Make transfer
3: Display the blockchain
4: Update the blockchain
q: Quit
Please input your option: 2


Please enter the username of payee:client1
Please enter the transfer amount:100
```

Figure 6: User Interface



```
1: Account information
2: Make transfer
3: Display the blockchain
4: Update the blockchain
q: Quit
Please input your option: 4


updating...
Update from 127.0.5.1
```

Figure 7: Update

4 Blockchain

4.1 Overview

After we set up the network and all the clients are signed in. We can process to the mission of blockchain. In this part, the first client will create the genesis block and it follows a creation of other blocks by the others authenticated clients. These blocks will be checked and added on the chain. The chain will become our account book. In using the hash function, we could store the important information of the block in a fixed length string(256 bits in this case), the string will be enormously different even if we change a little bit of the information, so furthermore we could know that whether its information is changed or lost.

4.2 Assumption

After discussing with professor about the requirement of this project, we simplified the original blockchain model and made assumptions as below:

- The initial assets of the blockchain is distributed manually.
- We assume that every client is online when a block is being transmitted.
- We assume that there is only one block being transmitted at one time.
- The chain only has one branch.

4.3 Genesis block

Everything has its beginning. In the case of blockchain, it's the creation of the genesis block. In our case, all the content in the block are 0. We make a hash of the content and get a string. We define the previous hash as 0.The genesis block is initialized in the blockchain of every node.

4.4 Normal block and transaction

Assume that user1 wants to transfer a certain amount of money to user2, the procedures are like this. User1 will first ask for the username of user2. Then by using the transfer option in the interface, user1 will input the username of the payee and the amount of money he or she would like to transfer. As shown in figure8. We set the information of username and transfer amount ordered and unchangeable through the code 'tuple() in python write this as comment below. Thus, a block is created and will be broadcast to all the nodes. The detail of the block is shown in figure9.

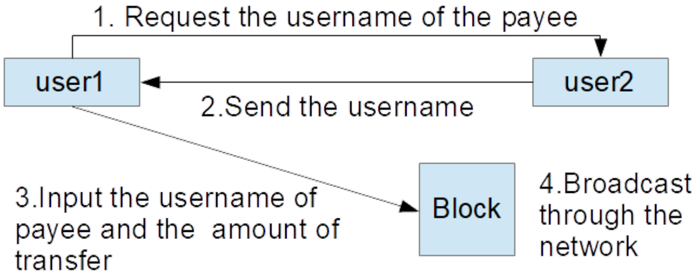
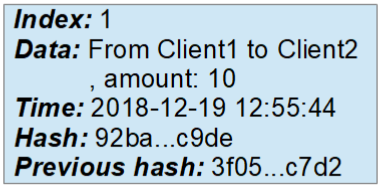

1. Request the username of the payee
user1
user2
2.Send the username
3.Input the username of payee and the amount of transfer
Block
4.Broadcast through the network

Figure 8: Make a transfer



Index: 1
Data: From Client1 to Client2 , amount: 10
Time: 2018-12-19 12:55:44
Hash: 92ba...c9de
Previous hash: 3f05...c7d2

Figure 9: The block

A short description of every message in the block is as follows:

- The index is the height of the block in the blockchain.
- The data includes the transmitter and the receiver of the transfer.
- The hash is a SHA-256 one taken from the content of the block.
- The previoushash is the hash of the previous block, which will be used to link the block to the chain.

4.5 Blockchain and verification

After the block reaches every node. By comparing the previoushash of current block to be same as the hash of the previous block, the new block can be linked to the blockchain. But before that, we also need to check if the payer has sufficient balance. As the current block contains the message of the source of the balance, all we need to do is to validate the source and check the payer hasn't transfer the money to another account. After this validation, the new chain is ordered are unchangeable. So the transfer is recognized and recorded by every node.

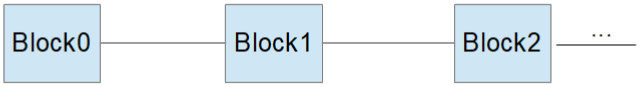


Figure 10: The blockchain

4.6 Mining

In the previous part, we discussed a lot about blockchain. This is just the frame of the bitcoin system. In order to add the transaction records into the blockchain, we need a process named as ”mining”. New transactions need to be included in a block with proof of mathematical workload to be validated. In real life, this proof is hard to generate because it can only be generated by trying billions of calculations per second. Miners need to run these calculations before their blocks are accepted and rewarded.

The mining is crucial for the security of the bitcoin system. It promises that we have enough balance for the transaction.

In this part, we add a concurrent mining process in our previous work as a proof of work of our blockchain network.

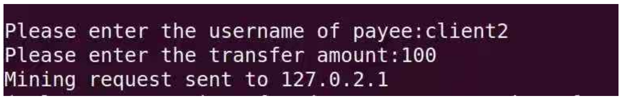
4.6.1 Assumption

The mining speed is determined by the difficulty. So in order to obtain the result in spending suitable time and simplify the condition, we assumed that:

- There is no incentive[1] for miners produced during the mining procedure, this is to make the transaction have single-input-single-output.
- To make the computation easier, we set a small degree of complexity: the resulting hash value begins with six '0'.

4.6.2 Solution

We add a supplemental element 'nounce' in the block as a variable. In order to start the mining process, a initial block is send to all nodes. The nodes will calculate the hashes of the blocks with different initial nounce values in utilizing loops. When a node gets a block with the hash starting with '000000', it sends a commit packet to all nodes, and the calculation processes in other nodes will stop. We get the related nounce, the related block is added into the chain the mining process is completed.
The client1 can send the mining request to client2, and the client2 can succesfully receive the request.



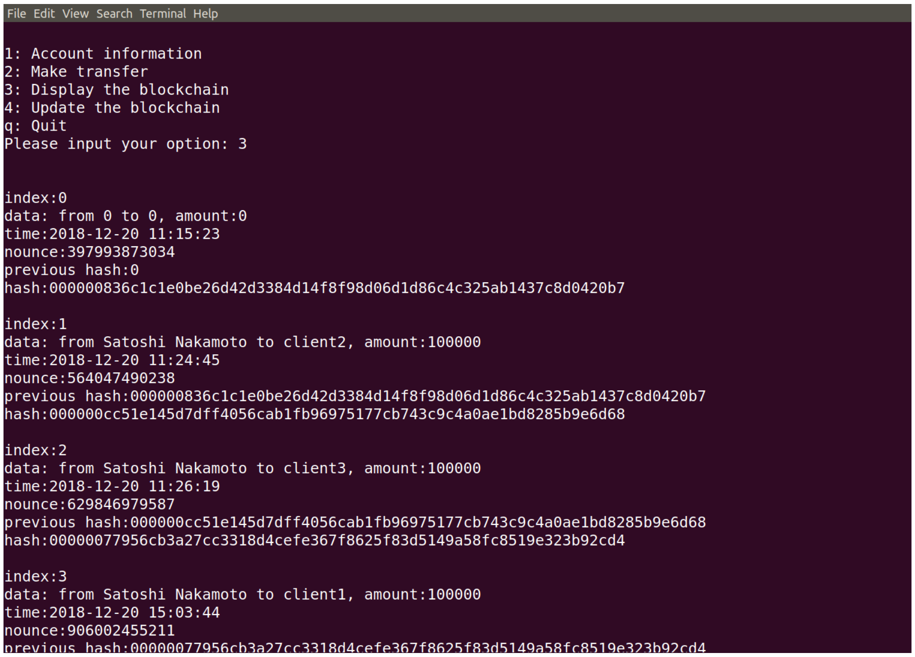
```
Please enter the username of payee:client2
Please enter the transfer amount:100
Mining request sent to 127.0.2.1
```

Figure 11: Client1 send the mining request

4.7 Result

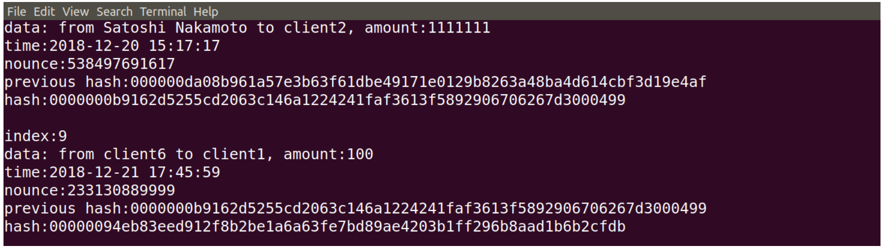
4.7.1 The blockchain

We can see the blockchain in figure12 and figure13. The index 0 is the genesis block. From index 1 we can see that ST make a 100000 amount transfer to Client1. The previoushash linked the index 1 to the genesis block, whereas the hash of index 1 will link with the previoushash in the next block. We can also see that the hash of each block starts with six zeros, which meets the requirement of mining.



```
1: Account information
2: Make transfer
3: Display the blockchain
4: Update the blockchain
q: Quit
Please input your option: 3


index:0
data: from 0 to 0, amount:0
time:2018-12-20 11:15:23
nounce:397993873034
previous hash:0
hash:000000836c1c1e0be26d42d3384d14f8f98d06d1d86c4c325ab1437c8d0420b7

index:1
data: from Satoshi Nakamoto to client2, amount:100000
time:2018-12-20 11:24:45
nounce:564047490238
previous hash:000000836c1c1e0be26d42d3384d14f8f98d06d1d86c4c325ab1437c8d0420b7
hash:000000cc51e145d7dff4056cab1fb96975177cb743c9c4a0ae1bd8285b9e6d68

index:2
data: from Satoshi Nakamoto to client3, amount:100000
time:2018-12-20 11:26:19
nounce:629846979587
previous hash:000000cc51e145d7dff4056cab1fb96975177cb743c9c4a0ae1bd8285b9e6d68
hash:00000077956cb3a27cc3318d4cefe367f8625f83d5149a58fc8519e323b92cd4

index:3
data: from Satoshi Nakamoto to client1, amount:100000
time:2018-12-20 15:03:44
nounce:906002455211
previous hash:00000077956cb3a27cc3318d4cefe367f8625f83d5149a58fc8519e323b92cd4
```

Figure 12: The Blockchain

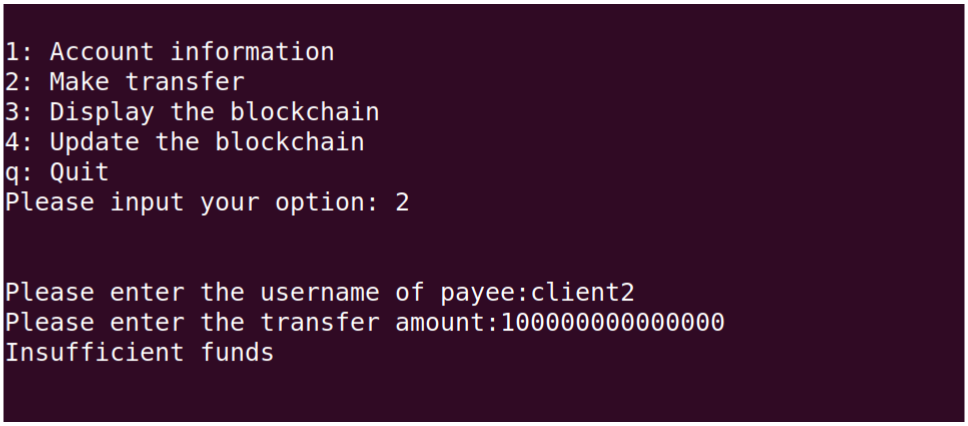


```
data: from Satoshi Nakamoto to client2, amount:1111111
time:2018-12-20 15:17:17
nounce:538497691617
previous hash:000000da08b961a57e3b63f61dbe49171e0129b8263a48ba4d614cbf3d19e4af
hash:0000000b9162d5255cd2063c146a1224241faf3613f5892906706267d3000499

index:9
data: from client6 to client1, amount:100
time:2018-12-21 17:45:59
nounce:233130889999
previous hash:0000000b9162d5255cd2063c146a1224241faf3613f5892906706267d3000499
hash:00000094eb83eed912f8b2be1a6a63fe7bd89ae4203b1ff296b8aad1b6b2cfdb
```

Figure 13: The Blockchain

4.7.2 Verification

If the balance of the client is insufficient for the transfer, the verification procedure will make sure that the block is not added to the blockchain. And the transaction is not recognizable.



```
1: Account information
2: Make transfer
3: Display the blockchain
4: Update the blockchain
q: Quit
Please input your option: 2


Please enter the username of payee:client2
Please enter the transfer amount:100000000000000
Insufficient funds
```

Figure 14: Insufficient Balance

5 Conclusion

In this project, we created a basic blockchain network of six users and one authentication center with mining function. Overall we faced many challenges, the most important being the fact that we had to do a lot on our own.

At first we have problem in understanding how much ”host.ini” and ”blockchain.dat” we should create in order to store the network and blockchain information. We read the requirement many times and make some discussions to decide that every node has one.

Second difficulty we met was trying to understand why we need to do the mining. Later on we realized that the mining is good for the security of the chain. It takes a lot of time to put the block on the chain, which means it cost time for the transaction to be recorded. In this way, it’s not easy for the user to inverse the transaction or generate false transactions to cheat others.

Third, we fogot to broadcast to all the node when we trying to do the broadcast of the block among users at first. Later on we corrected our mistakes.

We learned a lot about ”communication networks” in practice: code network topology, make broadcast among users and communication between user and center actually work, design good user interface, apply blockchain algorithm, understand and apply the mining, testing everything, etc. The most important one to us is a thorough understanding of the emerging technologies and its link with our theoretical courses and lab sessions.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.