COP5615: Distributed Operating Systems, Fall 2018

Project Report

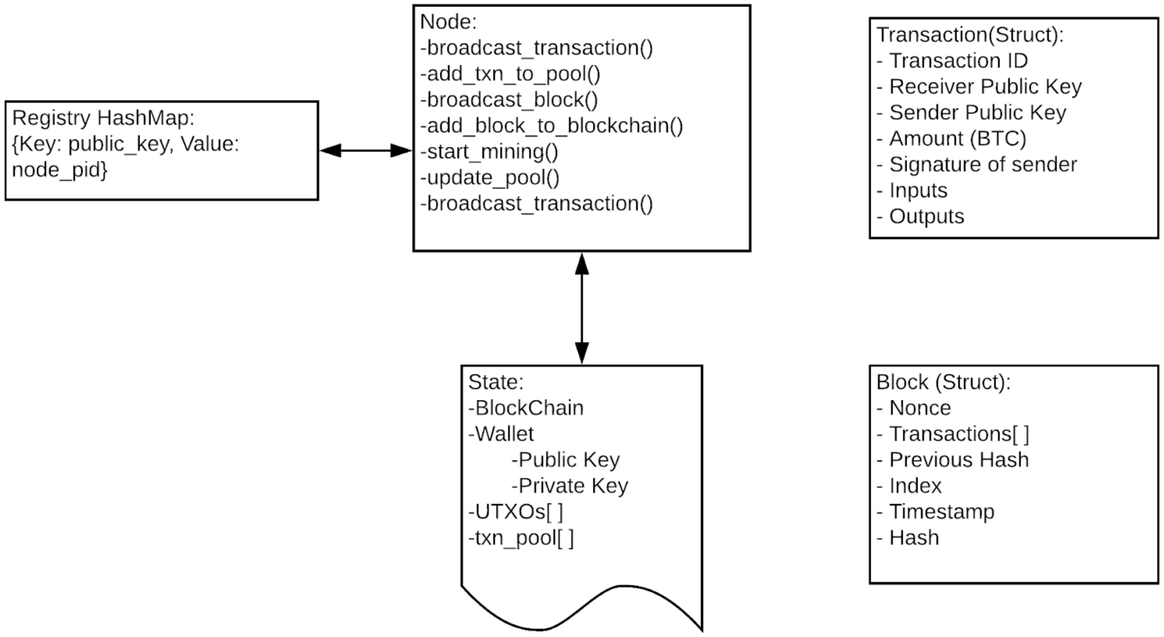
Bitcoin Simulator (Part 1)

Submitted by:
Kunwardeep Singh (UFID: 2421-3955)
Gayatri Behera (UFID: 3258-9909)

1. Description

In this project, blockchain-based distributed ledger and cryptocurrency protocol at a basic level is implemented. Here we discuss what parts of the Bitcoin protocol are implemented and what is working.

The high level view of the project structure is shown in this figure:



Following features are implemented:

1. Wallet is implemented as a struct containing public and private key stored in a node's state.
2. Each node has its own copy of the blockchain, transaction pool and the unspent transaction outputs (UTXOs) in its state.
3. Each node can broadcast transactions and blocks to every other node.
4. Every transaction has a signature which is generated by using private key of sender.
5. Transaction has inputs and outputs. Input will be the output of some previous transaction.
6. Nodes can verify if the transaction is correct by checking:
    a. If the signature is verified by using public key of the sender
    b. Check if inputs were valid
    c. Input amount equal to Output amount
7. Blocks are generated by adding all the transactions from the transaction pool of the miner and hash is calculated using correct Nonce. Difficulty is configurable and is set to 2.

8. Blocks are broadcasted by miners which are validated by the miners by checking:
    a. If the hash is correct using the set Nonce
    b. Number of zeroes in hash are equal to required difficulty
    c. Adding the block doesn't affect the blockchain

2. Test case scenarios

Following test scenarios are handled in the implementation:

1. Creation of genesis block:
    - Check if blockchain is created with the genesis block in first block
    - Verification of hash difficulty of the genesis block, which is set at 2

2. When new node joins the network:
    - Verify if the blockchain has been successfully sent to the new node
    - Verify if the UTXO copies are same in every node

3. Check balances:
    - Since UTXOs contain all unspent transactions of every node, adding all UTXO for a particular node will give the balance.

4. Transactions:
    - If the transaction amount is valid i.e. falls within the available UTXO range values for the sender node.
    - Verify the inputs and outputs of the transactions
    - If the broadcasted transaction exists in every node's transaction pool.
    - If a node tries to send amount greater than the UTXOs it has, it will be failed by all other nodes

5. Mining and broadcast of block:
    - Check if a coinbase transaction exists in the block
    - If the newly created block has the set hash equal to the calculated hash
    - If the newly created block has acceptable difficulty by checking number of zeroes
    - Consensus is taken by the network to ensure validity of a new block

6. Appending received block to the local blockchain:
    - Verify if the previous hash of the current block matches with the hash of the previous block
    - Verifying if subsequent blocks are being correctly appended by checking the indices
    - Re-calculate hashes of all blocks in the blockchain

7. Rewarding node on successful mining:
    - If reward is being presented to the node that has successfully mined the block and demonstrated POW.
    - Verify that the coinbase transaction amount is valid

8. Validation of the blockchain itself:
    - If for all blocks in the blockchain, the hash value using the nonce value is valid
    - If coinbase amount of a block is set to its expected value
    - If the previous hash of the current block matches with the hash of the previous block
    - Modifying transaction amount within some prior block in the blockchain, correctly causes blockchain to lose integrity, by failing any of the previous conditions.

3. Bonus Features

1. Both normal and coinbase transactions are implemented.On successful mining of a block, miner gets a reward amount equal to the coinbase transaction amount of that block.
2. Immutability of entire blockchain is handled by checking valid hashes, nonces and transactions in every block. If a transaction is changed in any of the block, subsequent blocks are invalidated.
3. Transactions and blocks are broadcasted just like in the Gossip protocol fully connected network.
4. Decentralization is achieved as every node has its own copy of blockchain, transaction pool and UTXOs.