Spring 2021, Lab 4 proposal
14736 Di Lu(dilu), Yuan Gu(guy)

Lab4 Project Proposal

Distributed Blockchain System

Brief introduction:

A blockchain project will allow peers to mine blocks and add to a peer-to-peer network.

The Project will follow the ideas in modern blockchain, using unique hashes to identify each block.

Basic: The user will be able to

1. mine a block,
2. read blockchain log from peers
3. broadcast its local blockchain to the peer-to-peer network.

Improvements:

If time permits, we will finish translation and real peer-to-peer networks.

1. **Translation** will allow peers to send and receive virtual asset from other peers;
2. **real peer-to-peer networks** will use Spark Web Framework instead of the transport lib in previous Lab to implement visible BlockChain

Project route map:

BASIC/checkpoint-I: Basic BlockChain in TransportLib

Model:

1. BLOCK:

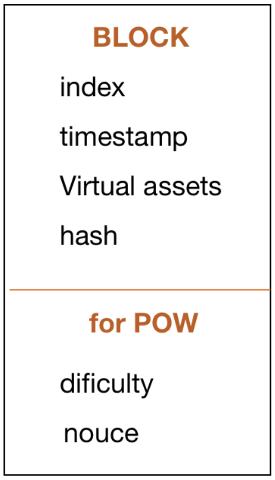
`Index` // the index in the entire chain
`timestamp`// the time the block is generated
`Virtual assets`// the 'money' data
`Hash` // through sha256 algorithm
`prevHash` // through sha256 algorithm, points to the previous hash

Spring 2021, Lab 4 proposal
14736 Di Lu(dilu), Yuan Gu(guy)

In current block design, proof of work (POW) should be added to a block. POW is introduced to increase the cost of appending a new block to the blockchain. The structure will be like:

Difficulty is an integer field indicating digits of leading zeros in a hash. The bigger the difficulty is, the harder to mine a block.

Nonce is a long number, which is used to recalculate the hash. If the new hash contains difficulty numbers of zeros, a block is mined.



2. The chain

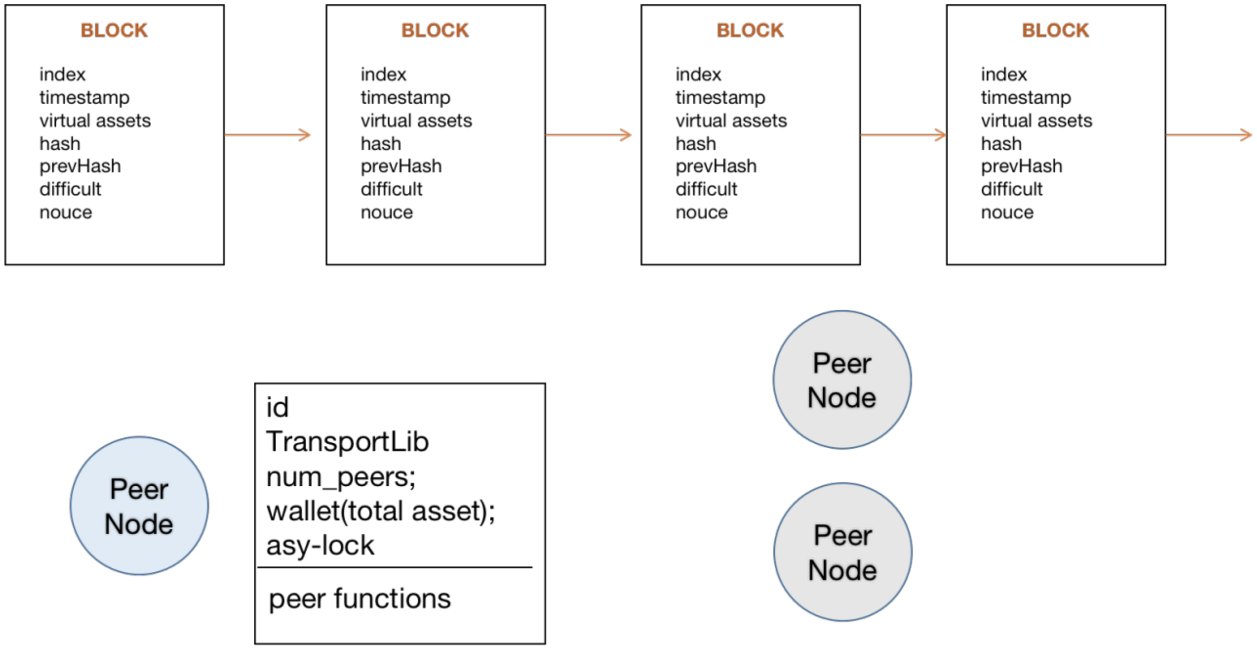
   The block chain can be maintained by a simple ArrayList:

   ```
   ArrayList<Block> blockChain
   ```

3. The Peer Node

Spring 2021, Lab 4 proposal
14736 Di Lu(dilu), Yuan Gu(guy)

Peer-to-peer global blockchain



Key Functions:

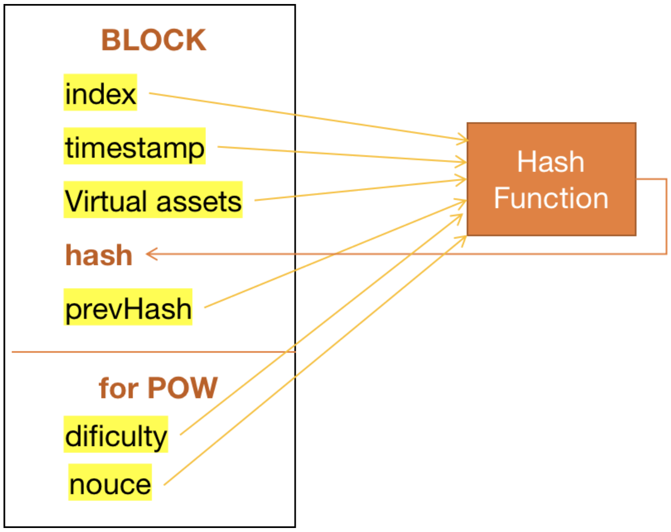
Some basic functions of a block chain should be:

Blockchain implements synchronous copy of ledger through p2p technology
Blockchain increases the cost of information dissemination and reduces the rate of information dissemination by increasing the Prove-of-work(pow)
Blockchain uses the length of the blockchain to judge the credibility of the data

1. Generate and mine Block:

A function to generate uniquely identifiable data according to the block data and maintain chain integrity.

Spring 2021, Lab 4 proposal
14736 Di Lu(dilu), Yuan Gu(guy)

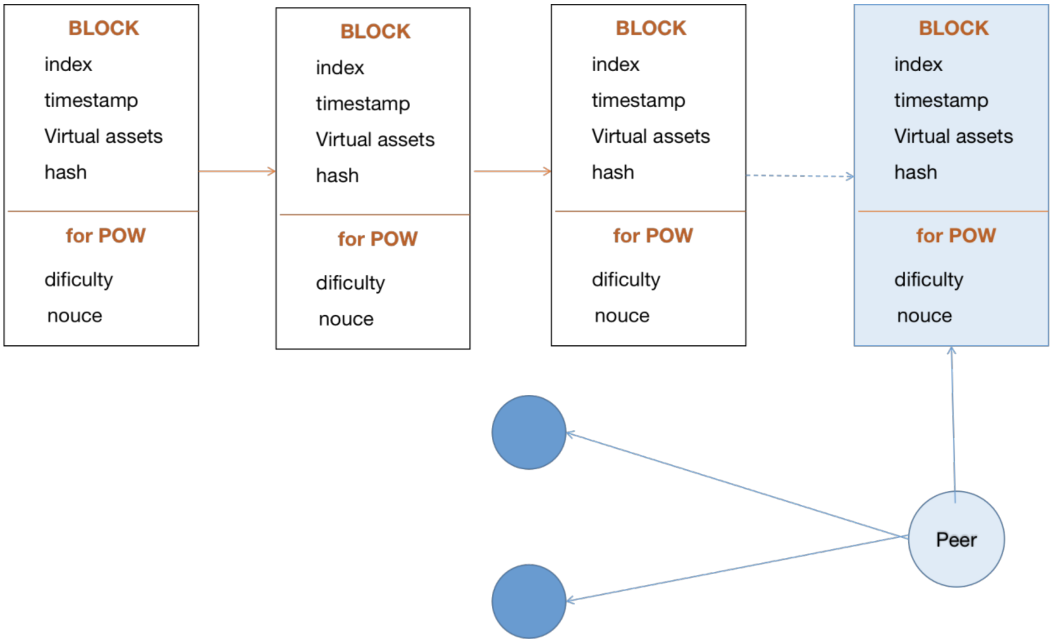


As to generating a block, it is often called ‘mining’.

To mine a block, we need to calculate hash based on all the information contained in the block, except the difficulty field. The work is done by enumerating all possible long `nonce` and calculating it’s hash, until it starts with a difficulty Number of Zeros.

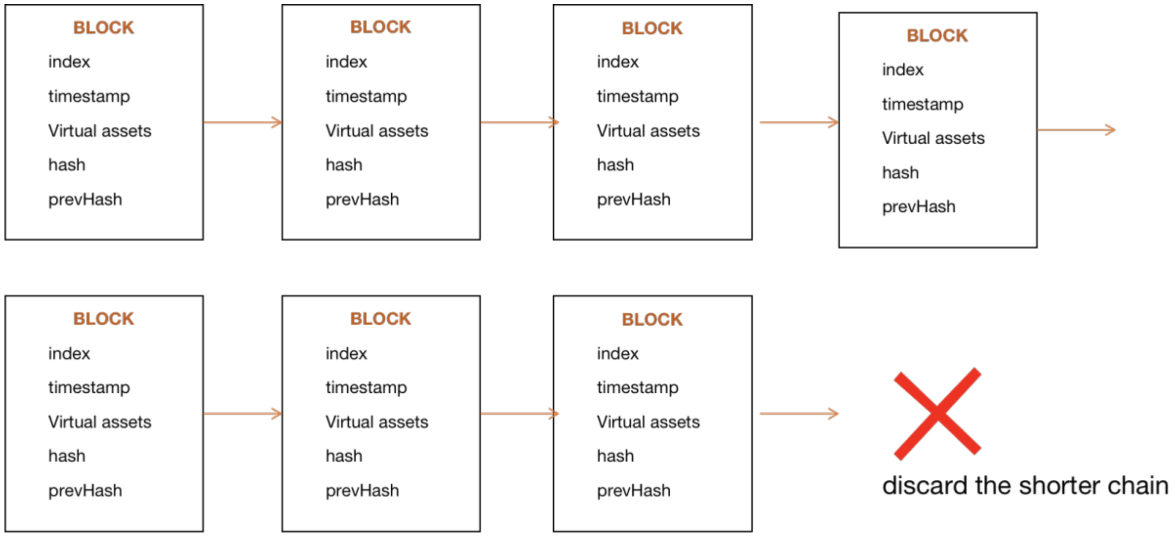
2. Validate blocks , Read and write blockchain

A peer should be able to read the current blockchain and apply to append new blocks to it. After a new block is mined, a user should be able to broadcast it to all peers in the peer network.



3. Global approve/discard conflict chains.

   As there may be many users trying to modify the blockchain, we should have schemas to keep the blockchain consistent and free of race conditions



Transport Layer:

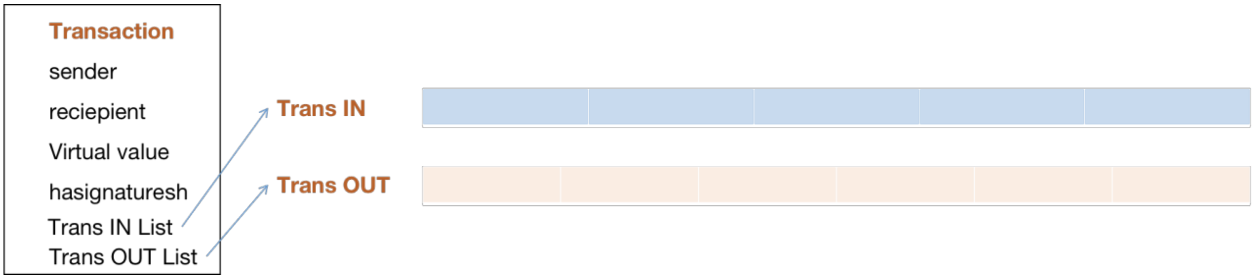
Basic: Transport Layer in previous lab(TransportLib)

If time permits: Transaction/checkpoint-II:

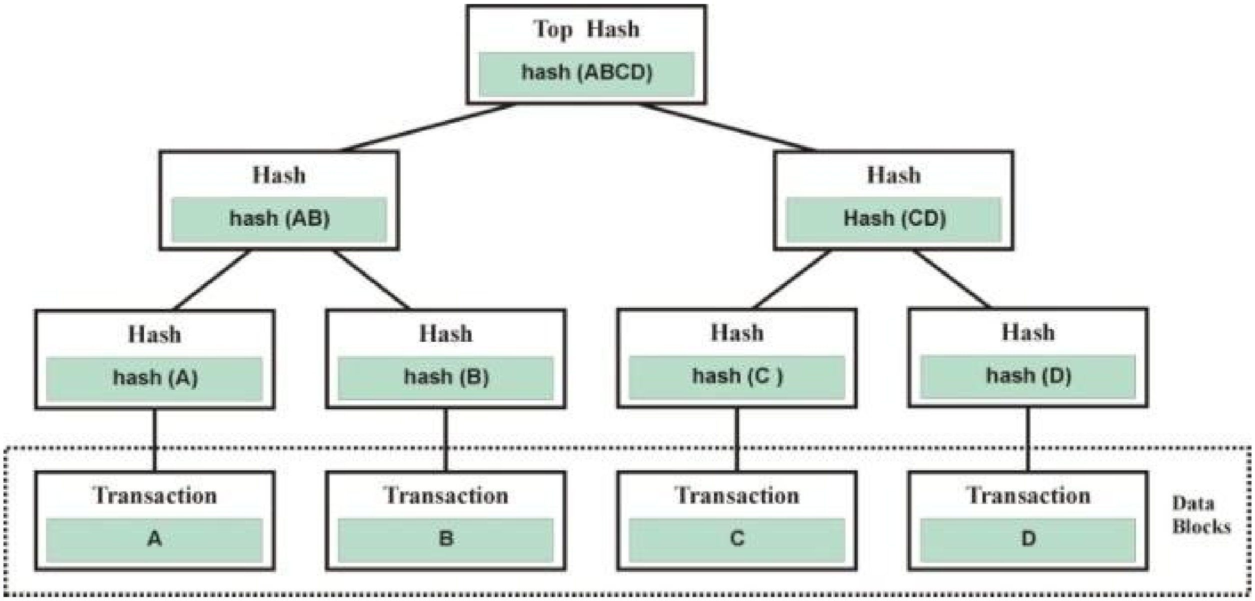
In a peer-to-peer network, a peer should be able to send money(virtual assets) to another. The sender and recipient should be the PublicKey of a peer.

Model

1. Transaction unit model



2. Merkle Tree(Hash Tree)

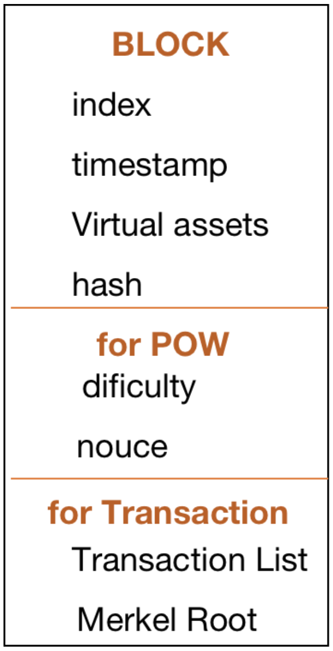


A Merkle tree is a binary-tree like hash tree structure.

It stores all the transactions in a block by producing a digital fingerprint of the entire set of transactions. It allows the user to verify whether a transaction can be included in a block or not.

Merkle Root is stored in the block header and makes the transaction tamper-proof.

3. The Block will have an extra field recording
`List<Transaction> transactions`

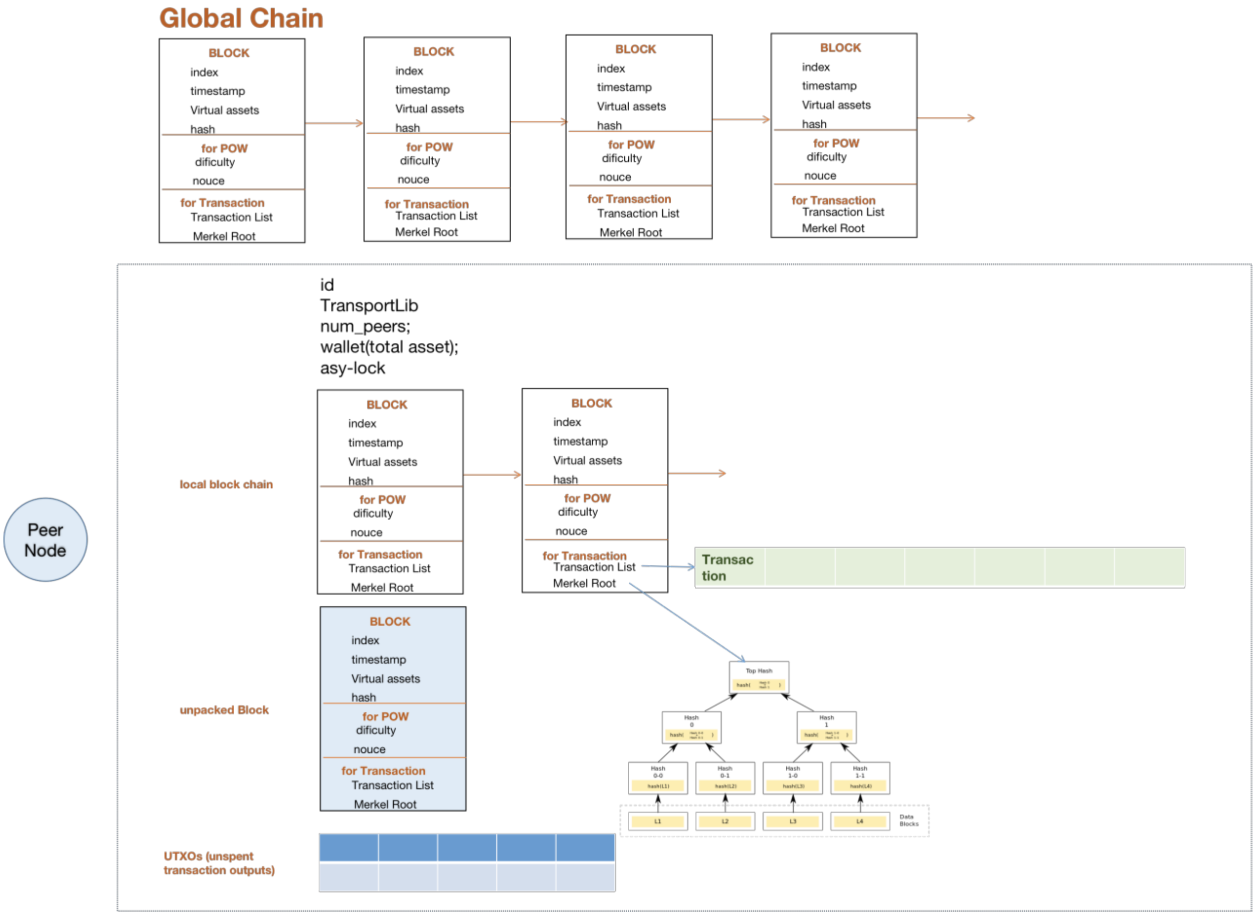


BLOCK
index
timestamp
Virtual assets
hash

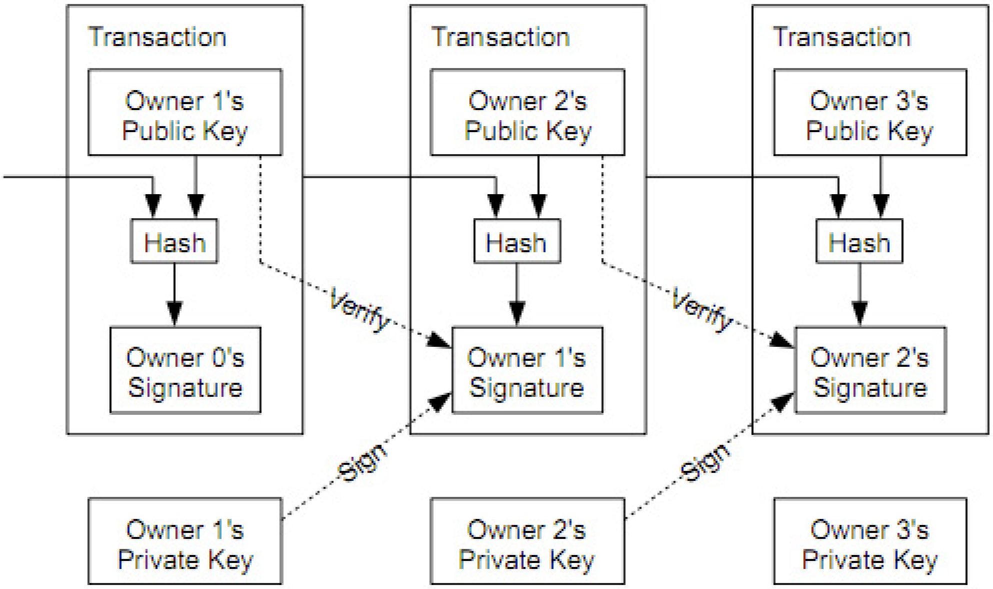
for POW
dificulty
nouce

for Transaction
Transaction List
Merkel Root

4. Peer Node



The Transaction verifying Process:



Improvements/final:

Real Transport Layer:

real communication: Spark Web Framework (if time permits)

- A verb (get, post, put, delete, head, trace, connect, options)
- A path (/hello, /users/:name)
- A callback (request, response) -> { }

  Ref: https://sparkjava.com/documentation#cookies

Points Breakdown on Coding Tasks:

Block mining/POW functionalities (data structures)	20
Broadcast, read, write, validation on blockchain (operations on data structures)	40
Schema that handles concurrency issues on operations and helps reach the consensus on decisions	40
Transportation Layer	20
Transaction verification and real p2p communication scheme	Bonus: 20 (each 10, if bonus is allowed)

(Disclaimer: we currently do not have a clear concept on the difficulty level of these subtasks. I generated these numbers based on my current experience. If instructors can see a more reasonable score/ratio distribution, please let us know. Thanks!)

Timeline:

Apr 19 - Apr 25:

1. Project build

2. Test Cases and makefile
3. Basic Implementation

Apr 26 - Apr 30:
Debug Basic

Apr 30 - May 3:
Project test
Implement transaction(or previous implementation/test)

May 4 - May 7:
Transaction test(or previous implementation/test)
Spark server test(or previous implementation/test)

May 8 6:59am:
due