## Design Document

### Members of Group 13

1. Nichit 2017MCS2089
2. Raj 2017MCS2098
3. Fai 2017MCS2525

### Language

Java

### Overview

**Master program**

* This will simulate the node creation and node failure
* Input
  + the number of nodes to be created
  + probability of node failure

**Nodes**

* Every node will use the same code
* Each node will communicate with each other using TCP connections, i.e. socket programming in Java

initTransaction()

listenTransaction()

broadcast()

broadcastReceive()

verifyDigitalSignature()

verifyTransaction()

getPublicKey()

### Protocol

We will use either of the below methods for communication, depending upon the implementation.

**Method 1:**

Each node will explicitly call the corresponding function directly, passing in the arguments as specified in the function.

**Method 2:**

Each node has a generic listener (server socket). The sender node will have a client socket for sending out messages. Based on the message header, the receiver will determine which internal function to call.

### Description of methods

|  |  |
| --- | --- |
| Method | Description |
| initTransaction() | Any node which wants to initiate a transaction will call this method |
| listenTransaction() | The receiver and the witness will decide whether to commit the transaction or not |
| Broadcast() | Transaction initiator will broadcast the transaction information to all online nodes |
| broadcastReceive() | All nodes will listen to the broadcast, verify the digital signature and input transactions. Once verified it will write the transaction to its ledger |
| verifyDigitalSignature() | It will verify the digital signature of the sender |
| verifyTransaction() | It will verify the input transactions with the node’s ledger |
| getPublicKey() | Get the public key from the distributed hash table. This might make further calls to other nodes |

### Some important class schemas

**Transaction class**

String transactionId

String senderId

String receiverId

String witnessId

Double amount

List<String> inputTransactions (storing transaction ID)

String digitalSignature

Bool valid

**Ledger class**

List<Transaction>

**Node class**

String nodeId

String publicKey

String ipAddress

Int port

**Distributed Hash Table class**

Map<String, Node> leafSet

Node[][] routingTable

Map<String, Node> neighborhoodSet

### Action plan

|  |  |
| --- | --- |
| Date | Action |
| Aug 6 – Aug 12 | * Implement P2P communication framework * Implement 2-phase commit protocol and broadcast methods |
| Aug 13 – Aug 19 | * Implement distributed hash table |
| Aug 20 – Aug 26 | * Implement virtual synchrony for concurrent broadcast * Check for double spend |
| Aug 27 – Sep 1 | * Final testing |