

# Final Report: The Presidency Protocol

## Blockchain Based Centralized Government

Hassib Ashouri, Jenil Thakker

May 13, 2019

### 1 Setup

For this project we implemented a simplistic version of the Delegated Proof-of-Stake protocol, by employing a voting mechanism. We setup our simulation with three clients and four delegates. The program chooses the top three candidates based on their received votes (configurable: we can change the number of "top" delegates). In this program, there are two important entities, namely the delegates and the government. The government maintains the received votes, selects the top three delegates, and determines the winning delegate who gets to add the block to the blockchain. The delegates proposes blocks and commits them once the winner has been selected from the top three. Lastly, the clients vote for their top three delegates.

In the simulation, figure 1 showing the output of the application, the clients make a total of two transactions (printed with updated balances), and vote for the top three delegates. The government then announces the winning delegate, after which the delegate commits the block to the blockchain. Lastly, the updated blockchain is printed to show the updated balances.

Note: The government charges, 0.9 JH-Coins as 0.01 as fees. The fees can be used to distribute rewards equally to the delegates.

### 2 Voting Rounds

The voting rounds represent cycles that the system goes through to process new transactions, take tax, distribute the rewards, and updates the ledger for all participants. Each voting round the clients who can vote picks chooses

```

Starting simulation. This may take a moment...
Initial balances to all accounts are
{
  '123': 100,
  '456': 200,
  '789': 100,
  'dell': 100,
  'dell2': 100,
  'dell3': 100,
  'gov': 1000 }
Alice has 100 coins.
Bob has 200 coins.
Charlie has 100 coins.

-----MAKING A TRANSACTION-----
Charlie(789): Made a transaction of 50 to 123

-----BEGIN VOTING-----
Alice(123): voted for dell
gov: one vote for dell
Bob(456): voted for dell
gov: one vote for dell
Charlie(789): voted for dell
gov: one vote for dell
gov: The top candidates for this round are dell,dell2,dell3

gov: dell is the WINNER to commit the new block!
dell: Committed a new block because the gov chose me
=> { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] }
dell2: Added block: { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] }
dell3: Added block: { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] }
dell4: Added block: { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] }
gov: Added 4.5 coins from taxes.
gov: distributed 0.45 of rewards.

-----PRINTING UPDATED BALANCES-----
{
  '123': 150,
  '456': 200,
  '789': 50.45,
  'dell': 100.81666666666667,
  'dell2': 100.81666666666667,
  'dell3': 100.81666666666667,
  'dell4': 100,
  'gov': 1004.5 }
-----PRINTING BALANCES-----
Alice now has 150 coins.
Bob has 200 coins.
Charlie now has 50.45 coins.
dell 1 now has 100 coins.
dell 2 now has 100 coins.
dell 3 now has 100 coins.

-----MAKING ANOTHER TRANSACTION-----
Alice(123): Made a transaction of 10 to 456

-----BEGIN VOTING: SECOND ROUND-----
Alice(123): voted for dell
gov: one vote for dell
Bob(456): voted for dell
gov: one vote for dell
Charlie(789): voted for dell
gov: one vote for dell
gov: The top candidates for this round are dell,dell2,dell3

gov: dell is the WINNER to commit the new block!
dell2: Committed a new block because the gov chose me
=> { "transaction": { "amount": 10, "tax": 0.8999999999999999, "fee": 0.01, "to": "456", "from": "123", "height": 2, "committer": "dell2", "prevBlock": { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] } } }
dell3: Added block: { "transactions": [ { "amount": 10, "tax": 0.8999999999999999, "fee": 0.01, "to": "456", "from": "123", "height": 2, "committer": "dell2", "prevBlock": { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] } } } }
dell4: Added block: { "transactions": [ { "amount": 10, "tax": 0.8999999999999999, "fee": 0.01, "to": "456", "from": "123", "height": 2, "committer": "dell2", "prevBlock": { "transactions": [ { "amount": 50, "tax": 4.5, "fee": 0.05, "to": "123", "from": "789", "height": 1, "committer": "dell" } ] } } } }

```

Figure 1: Expected output for the simulation.

a delegate to vote for and cast the vote the government. The government chooses the top 3 (configurable in the government class) delegates as candidates to propose blocks to be added to the blockchain.

The output of the application shows how in each round the winner commits its block then the rest of the candidates adds the new block to their blockchain. which is chained JSON object in our case for the sake of simplicity.

The end of the voting round shows the amount of tax the government collected and the amount of rewards that got distributed among the top 3 delegates. Also, it shows how the account balances changes as a result.