Developing a Distributed Distributed Consensus Protocol Consensus Protocol

Alec Grieser

Cryptocurrency Cabal, Final Project
University of Virginia — Class of 2017

December 2, 2015

Introduction

- Goal: Allow the network to agree to changes to the Bitcoin protocol
- Subproblems:
  - Protocol specification should be available in a clear, unambiguous way.
  - Nodes entering the network should be able to determine the protocol and update themselves.
  - Mechanism for changes to be proposed.
  - Network should be able to agree on changes to accept or decline.
- Addendum: The solutions to these should be as decentralized as possible.

Proposed Solution

- Specification: Modular specification mixing human- and machine-readable elements.
- Implementations include specification document.
- Checksum can be included in regular intervals in the coinbase parameter.
- Procedure for calling for a vote involving announcements to the blockchain.
- Two-stage secret vote using Bitcoin as votes.

Specification

- Comprehensive description of Bitcoin
  - Hashes used, block size, header details, transaction fields, script language, difficulty schedule, mining rewards, etc.
  - Exists unofficially currently in English: Bitcoin-Spec
  - Should include new information about voting procedures.
- Place in easy to digest form, e.g., JSON.
- Mix of English and formal mathematics
  - English allows for flexibility.
  - Mathematics allows for a clear and unambiguous specification.

Specification (cont.)

Example:

```
{
  "transaction": {
    "fields": {
       "inputs": {
         "description": "list of incoming txn_outputs",
          ...
        },
        "outputs": {
          "description": "list of outgoing txn_outputs",
          ...
        },
    },
    "max_size": 100000,
     ...
  },
  ...
}
```

Specification (cont.)

Example (cont.):

```
{
  "block": {
    "fields": {
       "block_header": {
           "fields": {
               "prev_block_header_hash": { ... },
               "merkle_root": { ... },
               "nonce": { ... },
               ...
           }
       },
       ...
    },
    "max_size": 10000000,
     ...
  },
  ...
}
```

Specification (cont.)

Example (cont.):

```
{
  "script": {
      "instructions": [
          {
              "word": "OP_DUP", "opcode": 118,
              "input" : "x", "output" : "x x",
              ...
          },
          {
              "word": "OP_HASH160", "opcode": 170,
              "input": "x",
              "output": "RIPEMD-160(SHA256(x))",
              ...
          },
          ...
      ],
      ...
  },
  ...
}
```

Client Updates

- Specification can be included in node source.
- Nodes can determine hash of own version of script.
- Checksum of script can be included in block header (up to 4 bytes).
- Block header information used to determine version to use with block.
- Nodes can use data in block chain to see need to upgrade.

Proposing and Accepting Upgrades

- Protocol includes specification for updating.
- Changes are proposed by members of the community.
- Anyone can call for a vote and anyone can vote.
- Proposals and voting are done by special transactions.
- Votes are initially secret and revealed after all votes are in.
- Bitcoin used as votes (proof of stake) and to propose vote.

Voting Proposal

- Use “hash puzzle” locking script.
- Create transaction with input size as vote.
- Place commitment in locking script to vote:

  ```
  OP_HASH256
  OP_DATA SHA256(SHA256(
      vote_id || specification_hash || nonce))
  OP_EQUALVERIFY
  ```

- Unlocking script (revealed in second stage):

  ```
  OP_DATA vote_id || specification_hash || nonce
  ```

- Problem?

Voting Proposal (cont.)

- Instead combine hash puzzle with standard P2PKH script.
- Locking script is then:

```
OP_HASH256
OP_DATA SHA256(SHA256(
    vote_id || specification_hash || nonce))
OP_EQUALVERIFY
OP_DUP
OP_HASH160
OP_DATA public_address
OP_EQUALVERIFY
OP_CHECKSIG
```

- Unlocking script:

```
OP_DATA signature
OP_DATA public_key
OP_DATA vote_id || specification_hash || nonce
```

Voting Proposal (cont.)

- Votes are kept secret until reveal.
- Values are stored on public ledger once revealed.
- Protocol should specify similar transactions for proposing elections.
- Time period should be fixed for voting (both first and second phase).

One Bitcoin, One Vote

- Nature of proposal means those with more money have more influence.
- Pros and cons:
  - Less "democratic."
  - Decreases spammers' influence.
  - Those with "stake" in system have say over its future.
- Anyone can leave at any time.

Conclusions

- Proposal would remove power from bitcoin developers.
- Puts decision in users (but possibly select few).
- Allows for system to evolve with common consent.
- Still concerns about whether miners would allow in all votes.