以太坊底层技术研究

(POA委员会选举机制)

|  |  |  |
| --- | --- | --- |
| 文件状态 | 当前版本 | *V1.0* |
| [√] 草稿  [ ] 正式发布 | 作 者 | 娄山林 李汝佳 段希楠 |
| 完成日期 |  |
| 文档模板 |  |
| 密 级 | *内部使用* |

变更历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 完成日期 | 变更记录 | 作者 | 审核 | ww批准 |
| V1.0 | 20171206 | 初稿 | 娄山林  李汝佳  段希楠 |  |  |
|  | 20171208 | 修改部分描述错误 | 娄山林 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1 名词介绍 4](#_Toc500338486)

[2 矿工投票方法 4](#_Toc500338487)

[3 委员会确定投票流程 4](#_Toc500338488)

[3.1 关键概念描述 4](#_Toc500338489)

[3.1.1 Epoch & checkpointInterval 4](#_Toc500338490)

[3.1.2 Snapshot 5](#_Toc500338491)

[3.2 投票方法 5](#_Toc500338492)

# 名词介绍

节点：普通的以太坊节点，没有区块生成的权利。

矿工：具有区块生成权利的以太坊节点

委员会：所有矿工的集合

# 矿工投票方法

* + - * 用户通过RPC接口，调用Propose(address common.Address, auth bool)方法（consensus/clique/api.go）,进行投票，address表示要投票的节点的地址，auth表示要从将该地址加入委员会，还是从委员会中删除。
      * Propose方法将address和auth两个输入参数写入到clique.proposals集合中。
      * 任何一个委员会的委员，可以在任意时刻进行投票，投票包括两种，即加入委员会和从委员会中删除。

# 委员会确定投票流程

**type** Clique **struct** {

config \*params.CliqueConfig // Consensus engine configuration parameters

db ethdb.Database // Database to store and retrieve snapshot checkpoints

recents \*lru.ARCCache // Snapshots for recent block to speed up reorgs

signatures \*lru.ARCCache // Signatures of recent blocks to speed up mining

proposals **map**[common.Address]bool // Current list of proposals we are pushing

signer common.Address // Ethereum address of the signing key

signFn SignerFn // Signer function to authorize hashes with

lock sync.RWMutex // Protects the signer fields

}

## 关键概念描述

### Epoch & checkpointInterval

* CheckpointInterval：为常量1024（consensus/clique/clique.go中定义），即每当区块链的高度为1024的整数倍时，到达checkpointInterval时间点。
* Epoch：默认为30000（cmd/puppet/wizard\_genesis.go中makeGenesis方法中定义），即每当区块链的高度为30000的整数倍时，到达Epoch时间点。

### Snapshot

Snapshot是一个快照，矿工程序在区块链高度为CheckpointInterval的整数倍时，会

对当前相关数据和状态形成快照，并存储到数据库中

snapshot结构体（consensus/clique/snapshot.go）关键成员：

* Number：生成快照时的区块链高度
* Signers：生成快照时的委员会地址
* Votes：生成快照时所有的投票集合
* Tally：被投票的节点集合，其中的Tally是该节点被投票的次数

**type** Snapshot **struct** {

config \*params.CliqueConfig // Consensus engine parameters to fine tune behavior

sigcache \*lru.ARCCache // Cache of recent block signatures to speed up ecrecover

Number uint64 `json:"number"` // Block number where the snapshot was created

Hash common.Hash `json:"hash"` // Block hash where the snapshot was created

Signers **map**[common.Address]**struct**{} `json:"signers"` // Set of authorized signers at this moment

Recents **map**[uint64]common.Address `json:"recents"` // Set of recent signers for spam protections

Votes []\*Vote `json:"votes"` // List of votes cast in chronological order

Tally **map**[common.Address]Tally `json:"tally"` // Current vote tally to avoid recalculating

}

## 投票方法

所有投票都是在委员生成新区块的过程中完成，具体流程如下：

1. 委员生成新区块时，先为该区块初始化一个header。*（prepare方法，consensus/clique/clique.go）*
2. 从proposals中随机获取一个投票，将被投票的节点地址写入header.coinbase，将提名是添加还是删除写入header.Nonce(添加：0xffffffffffffffff 删除：0)，若该委员生成的这个区块最终被写入区块链，则header中的投票也被写入区块链。*（prepare方法，consensus/clique/clique.go）*
3. 委员在生成新区块时，会创建新的snapshot，新的snapshot是由上一checkponitinterval时间点存储到数据库中的快照加入当前时间点和checkpointinterval时间点之间所有的headers数据组成。添加headr过程中，若该header的number是Epoch时间点，则会将snap中的Votes和Tally两个集合清零。

*（apply方法，consensus/clique/snapshot.go）*

1. 新的snapshot添加headr过程中，会检查每一个header中存储的投票，若该投票snap.Votes中已经存在，则将snap.Votes和snap.Tally两个集合的该投票删除。

*（apply方法，consensus/clique/snapshot.go）*

将每一个header中有效的提名写入新snapshot的snap.Votes和snap.Tally集合。*（apply方法，consensus/clique/snapshot.go*）

1. 判断snap.Tally集合中某个被提名的节点，提名的次数是否大于snap.Signers的1/2,即是否有超过一半的委员对该节点进行投票，若超过，则投票成功，该节点会被添加到委员会或者从委员会中删除。*（apply方法，consensus/clique/snapshot.go）*

注释：snapshot快照中的记录的委员会，即Signers集合，初始化时来源于创世块header中的Extra