# 验证主节点共识算法设计-V0.0

## 前言

简单的流程和功能设计，仅供6.30版本软件框架设计。

## 方案选择

在Matrix中，验证主节点主要完成如下服务：

1. 数据的打包（交易的共识）；
2. 分叉的鉴定（区块的共识）；
3. 区块的鉴定（区块的共识）；
4. 广播/网络刷新生成（特殊区块的共识）

区块链的共识主要在验证者主节点完成，为了代表大部分权益者的利益，采用DPOS共识实现。

Miner采用POW共识，为区块链增加工作量证明，防篡改，同时为整个链提供时钟驱动。

## 共识使用位置

DPOS共识主要在三个部分体现：

1. 验证者间进行共识，生成共识信息，发送到矿工主节点，放入区块中；
2. 矿工主节点对决议信息执行DPOS共识验证，合法后通过区块广播；
3. 验证者接收矿工发出的区块，按照规则执行验证，投票表决区块的合法性，通过控制信息打包入下一个区块。

## 共识流程

1. leader的*选取*；
2. 数据同步；
3. 投票；
4. 生成决议信息；

### Leader的选取

考虑有两个方法：

1. ***按照某种顺序对验证主节点进行排序，轮流当选Leader。***

优点：简单，抑制部分Leader节点作恶，导致网络长时不出块等异常；

缺点：承担验证者获取的回报难于公平分配（按照下注越多，收益越多的原则，如果轮流出块且分配符合该原则，需要知道其他交易者的地址信息，不知道能否实现）

1. ***在选举周期开始，根据数学方法生成这样的一个出块序列：使得一个验证者出块概率和抵押值呈单调递增关系，一个验证者尽量不要连续出块。***

优点：分配简单，符合下注越多，收益越多的原则；

缺点：投入高的节点，一旦作恶/离线，对网络影响较大。

**目前可以先按照第一种实现。**

### 验证者共识流程

每一条请求消息处理结果的应答，作为请求消息的ACK。在这个过程中需要考虑超时和重传。

假设网络传输时延为T1，数据传输时延为：数据包大小L/传输速率R。各节点交易量为平均交易量。假设我们的系统需要支持1000TPS/s，每个交易256Bytes，效率50%。按照12s出一个块计算，由于每个节点交易数据采用双备份机制，那么每个验证节点本地交易大小为1000 \* 256 \* 8 \* 2 \* 2 \* 12 / 11= 8937Kb。

第一个会话时间：2T1 + 89370Kb/R。假设R为100Mbps那么会话时延为2T1 + 88M/100M/s。假设T1为0.5s。那么第一次会话时延为1.9s。

第二次会话，Leader向Follower发送全部交易信息，Follower向Leader发送验证信息（数据量相对比较少，忽略），整个会话时延和第一次一样，约为1.9s。

第三次会话，Leader向Follower发送重新打包的Bitmap，Follower向Leader发送签名信息数据量都比较小，整体时延为2T1计算，约为1秒钟。

那么三个阶段时延共为5s。如果按照12s出块速度计，最多支持2次重传。或者，整体不能超过12s。

### 打包数据验证内容

考虑验证如下内容：

1. 每个交易的合法性，按照交易的验证方法；

### 区块鉴定的合法性

考虑验证如下内容：

1. 和区块链验证一样，交易费，头等验证
2. 对所有交易验证；
3. 验证区块中是否完整包含上次验证者的打包数据；
4. 验证打包数据中签名是否合法，权益是否合法；

### 分叉鉴定

先不实现。

### 广播/网络涮新信息共识

作为特殊交易进行共识，及投票选举。后续出细节设计。

### 作弊手段及处理方式分析

#### 按照数据流分析。

两个大的数据流：

矿工不出块：矿工采用POW机制，除非所有矿工均不出块。矿工通过出块赚取收益，不会做这种损人不利己的行为。