Matrix交易验证与共识(V1.0)

1. 上层节点交易洪泛(矿工验证者通用)(2018-6-28)
   1. 第一步，发送者发送交易签名中的S值及编号N，打包发送给接收者
   2. 第二步，接收者收到交易后，检查交易是否存在，如果全部存在则结束
   3. 第三步，接收者将缺少的交易编号N打包发送给发送者
   4. 第四步，发送者将上一步请求的交易数据去掉S，加上编号N，发送给请求的接收者
   5. 查找排序功能
2. 交易编号设计
   1. 由于负载均衡的需要，每秒每个节点最大发送200Tx，20个区块后没有验证通过的交易会被丢弃，则编号生存最大数量为200\*15\*20=54000
   2. 验证节点和矿工节点顶级节点最大个数为128个
   3. 用3byte完成上面的编号设计，节点编号共128个，交易序号共131072个

00000000 1111111 1 11111111 11111111

交易序号共131072个

节点编号共128个

* 1. 矿工和验证者每个节点分配一个唯一的节点编号，可用nodeId排序编号实现
  2. 每个节点发送交易的编号从0开始累加，每次加1，已有编号的交易不重新编号
  3. 编号发生重复时，旧交易已经丢弃，小概率没有丢弃的，S值也不相同，修改编号下的交易就行

1. 交易验证
   1. 每个验证者发送交易验证，按照交易洪泛的格式。每个验证者需要去重，如果其他节点已存在交易，则只发编号即可
      1. 查找
   2. 验证者接到编号，发现没有该编号的交易，可请求发送者发送交易
   3. 验证者需要保存之前20个块区间内所有未上链交易以及3个块之内的上链交易详细信息（交易和编号）
      1. 时间戳
   4. 删除：如果收到满足共识验证者验证失败的回复，本交易则删除，并广播交易删除消息，包含交易编号及验证者签名，否则等够时间再删除
   5. 超过3个块后，3个块前的块上交易即可删除
   6. 交易验证的签名消息为验证成功的消息编号打包，及对消息编号hash的签名，共（N\*3+65）Byte，
   7. 压缩和加密，加个标志，p2p去做
2. 区块验证
   1. 区块验证由Leader发起，包括区块头及交易编号集：
      1. 区块头： 544 (492+3bigInt+extra)我们需要发送500左右
      2. 1000TPS\*15m\*3byte\*8 = 360K
      3. Leader的签名：65Byte
      4. 发送10个节点大概3.6M + 45K = 3.645M
   2. 验证返回签名对区块头已有信息的hash+true/false的签名
      1. Var TRUE = 0; FALSE = 1; // true = 0 保证true的时候为原签名
      2. r = k\*G //k为随机数，G为椭圆曲线基点
      3. s = [(hash+ture / false) + r.x\*p]/k //p为私钥
      4. v = r.y % 2 + (ture/false)\*2 //1byte
      5. 签名消息为[r.x,s,v] 共65byte
   3. Timeout时没发消息的Validator为验证失败
   4. Miner只需打包验证成功的签名
3. 出块失败共识
   1. Leader接到最新块后广播心跳消息，等待时间为1秒
   2. Leader广播心跳消息后，广播新区块验证消息，等待时间为0.5秒
   3. Validator接到心跳消息，则等0.5秒超时，否则等1.5秒超时
   4. Leader验证签名没有得到足够的签名或失败签名，需要等待1秒，超时[t0,t0+1]
   5. 下一个Leader优先发起出块失败共识，等待时间为2秒[t0+1,t0+3] [t0+1.4,t0+3.4] t1[t0+0.6,t0+2.4]
   6. 出块失败共识未达成，则Leader等1秒再次发起共识 [t1,t1+3]
   7. 出块失败共识达成，则回到b)。
   8. 如果e)超时，则顺推下一个leader发起两个出块失败的共识，依次类推（t0+3）（t1+3）
   9. 出块共识消息包括
      1. Leader差距值：发起Leader高度-出块失败Leader高度
      2. 等待开始消息(上一个区块出块、心跳、新区快验证)
      3. 以上消息的签名
   10. 其他Validator需要验证
       1. 等待开始消息是否正确
       2. 时间间隔是否正确
       3. 是否满足出块失败的条件
   11. 签名消息包括
       1. 上一个消息的hash+true/false
       2. 计算方法同区块签名验证
       3. 如果是区块验证签名时的等待，如果收集了满足共识的签名，则附上自己收集区块正确的签名
   12. 出块失败共识 会不会死锁，【通过，不通过，没有达成共识】
       1. 首先要满足拜占庭问题的前提，即2/3以上诚实节点
       2. TimeOut的问题，多等一轮就达成共识了
       3. 区块验签没有达到2/3而等待，就是Timeout问题
       4. 区块验签达到2/3，共识失败，肯定有Validator发给矿工挖矿了，为了保证不再次发起出块失败共识，发给矿工挖矿的Validator需要签名的时候附上自己收集区块正确的签名
4. 矿工出块
   1. 矿工出块目前为POW算法
   2. 每个Validator都可以给矿工发送挖矿消息，为区块头：以太坊原有区块头(500) + Validator正确签名(65\*11)
   3. 矿工挖矿成功后，将nonce，coinBase，blockHash传给Validator，共(8+20+32)=60
   4. 基金会布置备份矿工，当矿工在指定时间内挖不出矿时间(20s)，Validator可以选择备份矿工的挖矿结果。备份矿工与普通矿工同时挖矿，但是不收矿工费
   5. 为了保证挖矿时间，每20秒后，挖矿难度折半
   6. 矿工可提供当前难度和折半难度，备份矿工不需要等待，Validator决定使用数据
5. 网络数据流量统计
   1. 单节点200TPS，Validator：11，Miner：32，Miner交易重复率80%，Validator:20%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阶段 | 发送 | 接收 | 总计 | 合计 |
| Validator 交换交易数据 | 573440 | 573440 | 1146880 | 6932480 |
| 38400 | 38400 | 76800 |
| 2854400 | 2854400 | 5708800 |
| 矿工广播交易阶段 | 1835008 |  | 1835008 | 4149248 |
|  | 30720 | 30720 |
| 2283520 |  | 2283520 |
| 钱包发送交易 | 403200 | 51200 | 454400 | 454400 |
| Validator验证交易 | 53200 | 53200 | 106400 | 106400 |
| 区块验证阶段 | 3640000 | 5200 | 3645200 | 3645200 |
| 发送给矿工挖矿 | 311040 | 10240 | 321280 | 321280 |
| 区块验证瞬间流量总计 | 11992208 | 3616800 | 15609008 | 15609008 |
| 非验证期间流量 | 8041168 | 3601360 | 11642528 | 11642528 |

* 1. 单节点100TPS，Validator：11，Miner：32，Miner交易重复率80%，Validator:20%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阶段 | 发送 | 接收 | 总计 | 合计 |
| Validator 交换交易数据 | 317440 | 317440 | 634880 | 3527680 |
| 19200 | 19200 | 38400 |
| 1427200 | 1427200 | 2854400 |
| 矿工广播交易阶段 | 1015808 |  | 1015808 | 2172928 |
|  | 15360 | 15360 |
| 1141760 |  | 1141760 |
| 钱包发送交易 | 201600 | 25600 | 227200 | 227200 |
| Validator验证交易 | 29200 | 29200 | 58400 | 58400 |
| 区块验证阶段 | 3640000 | 5200 | 3645200 | 3645200 |
| 发送给矿工挖矿 | 311040 | 10240 | 321280 | 321280 |
| 区块验证瞬间流量总计 | 8103248 | 1849440 | 9952688 | 9952688 |
| 非验证期间流量 | 4152208 | 1834000 | 5986208 | 5986208 |

* 1. 不同状态下网络流量对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tps | V-R | M-R | Vnum | Mnum | Max Data | Av Data |
| 200 | 0.2 | 0.8 | 11 | 32 | 15609008 | 11642528 |
| 200 | 0.2 | 0.5 | 11 | 32 | 19080368 | 15113888 |
| 200 | 0 | 0 | 11 | 32 | 26312368 | 22345888 |
| 100 | 0.2 | 0.8 | 11 | 32 | 9952688 | 5986208 |
| 100 | 0.2 | 0.5 | 11 | 32 | 11688368 | 7721888 |
| 100 | 0 | 0 | 11 | 32 | 15304368 | 11337888 |
| 200 | 0.2 | 0.8 | 11 | 21 | 14075784 | 10216224 |
| 200 | 0.2 | 0.5 | 11 | 21 | 16353864 | 12494304 |
| 200 | 0 | 0 | 11 | 21 | 21597064 | 17737504 |
| 100 | 0.2 | 0.8 | 11 | 21 | 9098824 | 5239264 |
| 100 | 0.2 | 0.5 | 11 | 21 | 10237864 | 6378304 |
| 100 | 0 | 0 | 11 | 21 | 12859464 | 8999904 |

TPS影响最大