测试样例（部分）

# 区块验证服务测试用例

## 测试样例

1. 区块验证服务身份变更消息-不同的当前身份下收到身份变更消息

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 | 测试身份消息的处理逻辑 |
| 测试环境 | 一台pc机器 |
| 预置条件 | 无 |
| 测试过程 | 块高递增依次输入各种身份，调用handleRoleUpdatedMsg处理。通过对 processManage中process的数量及状态判断结果 |
| 预期结果 | 1. 测试函数正常执行，没有挂起等异常信息 2. 每次输入，会删除比输入高度小的process 3. curProcess的设置正常，process身份设置成功 4. 身份是验证者和广播节点时，process状态由idle转为运行状态 |

1. 区块验证服务leader变更消息-非验证者身份下，对各高度leader变更消息的处理

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 | 测试leader变更消息的处理逻辑 |
| 测试环境 | 一台pc机器 |
| 预置条件 | 先输入一个高度为10，身份为矿工的CA身份消息 |
| 测试过程 | 按不同块高输入leader变更消息，通过对 processManage中process的数量及状态判断结果 |
| 预期结果 | 1. 小于10高度的leader消息，不会创建process，丢弃消息不处理 2. 大于12高度的leader消息，不会创建process，丢弃消息不处理 3. 高度10的leader消息，对process设置leader成功。但是状态为idle不运行 4. 高度11、12的leader消息，创建process，设置leader成功，不运行 |

1. 区块验证服务leader变更消息-验证者身份下，对各高度leader变更消息的处理

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 | 测试验证者身份下，对leader变更消息的处理逻辑 |
| 测试环境 | 一台pc机器 |
| 预置条件 | 先输入一个高度为10，身份为验证的CA身份消息 |
| 测试过程 | 按不同块高输入leader变更消息，通过对 processManage中process的数量及状态判断结果 |
| 预期结果 | 1. 小于10高度的leader消息，不会创建process，丢弃消息不处理 2. 大于12高度的leader消息，不会创建process，丢弃消息不处理 3. 高度10的leader消息，对process设置leader成功。并进入下一步流程，查询leader的req请求 4. 高度11、12的leader消息，创建process，设置leader成功，不运行 |

1. 区块验证服务leader变更消息-验证者身份下收到正确的区块验证请求

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 | 测试正常流程下，收到正确的区块验证请求流程 |
| 测试环境 | 一台pc机器 |
| 预置条件 | 无 |
| 测试过程 | 在block\_verify\_test.go中运行TestRightRequestMsg\_01测试函数  创建eth对象及签名账户，然后依次输入身份(验证者)， leader变更消息和正确的区块验证请求消息。然后检查当前process情况，判断状态是否为等待dpos阶段状态 |
| 预期结果 | 1. 测试函数正常执行，没有挂起等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 区块验证服务leader变更消息-正确投票输入流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 无 |
| 测试过程 | 在block\_verify\_test.go中运行 TestVoteMsg\_01测试函数  创建eth对象及签名账户，然后依次输入身份(验证者)， leader变更消息和正确的区块验证请求消息。设置3验证者模式，并模拟输入其他2个验证者的同意投票。处理投票请求后，检查当前process情况，判断状态是否为结束状态，并结束结果消息 |
| 预期结果 | 1. 测试函数正常执行，没有挂起等异常信息 2. 测试函数正常执行，测试结果为pass |

# DPOS共识引擎测试用例

## 测试样例

1. 3验证者情况下，输入全同意签名列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 3验证者 |
| 测试过程 | 创建模拟数据，3个验证者。并用他们的私钥对Hash1进行签名。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash1和签名列表，看DPOS是否通过。再次调用接口，输入Hash2和签名列表，看DPOS是否返回签名数量不够错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 2验证者情况下，输入全同意签名列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 2验证者 |
| 测试过程 | 创建模拟数据，2个验证者。并用他们的私钥对Hash进行签名。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表，看DPOS是否返回股权者数量不够的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 3验证者情况下，输入含1个不同意签名列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 3验证者 |
| 测试过程 | 创建模拟数据，3个验证者。并用他们的私钥对Hash进行签名，其中一个签名为不同意签名。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否返回不同意签名数量过多的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 7验证者情况下，输入含6个签名的列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 7验证者 |
| 测试过程 | 创建模拟数据，7个验证者。并用6个私钥对Hash进行签名。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否返回签名数量不够的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 7验证者情况下，输入含7签名的列表流程，其中两个签名相同

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 7验证者 |
| 测试过程 | 创建模拟数据，7个验证者。并用6个私钥对Hash进行签名，并随机复制一个签名组成签名列表。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否返回签名数量不够的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 11验证者等股权情况下，输入含8个签名的列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 11验证者，股权相同 |
| 测试过程 | 创建模拟数据，11个验证者。并用8个私钥对Hash进行签名，组成签名列表。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否返回签名股权不够的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 11验证者不等股权情况下，输入含8个签名的列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 11验证者，其中一个10股权，其他验证者3股权 |
| 测试过程 | 创建模拟数据，11个验证者。并用8个私钥(10股权在其中)对Hash进行签名，组成签名列表。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否通过 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |

1. 11验证者不等股权情况下，输入含11个签名的列表流程

|  |  |
| --- | --- |
| 测试项目 |  |
| 用例编号 |  |
| 用例名称 |  |
| 用例目的 |  |
| 测试环境 | 一台pc机器 |
| 预置条件 | 11验证者，其中一个20股权，其他验证者3股权 |
| 测试过程 | 创建模拟数据，11个验证者。并用全部私钥对Hash进行签名，其中20股权的签名不同意，组成签名列表。然后调用DPOS共识引擎的 VerifyHashWithNumber接口输入Hash和签名列表。看DPOS是否返回不同意股权太多的错误 |
| 预期结果 | 1. 测试函数正常执行，没有panic等异常信息 2. 测试函数正常执行，测试结果为pass |