**接口设计总结**

# 文档说明

本文档用来总结0730各个模块之间的消息事件接口和方法接口

消息事件接口：消息接口（提供给别人的和别人提供给自己的都要写）

* 消息名称：
* 消息来源模块：
* 消息使用模块：
* 结构体：注意各成员变量的值域，注意双方必须沟通确定
* 接口方式：直接传结构体还是经过marsha发出
* 备注：

方法接口：模块间函数直接调用（提供给别人的和别人提供各自己的都要写）

* 同上

目录

**[1.矿工服务](#_Toc6484_WPSOffice_Level1)** **[1](#_Toc6484_WPSOffice_Level1)**

**[2. 随机数生成服务](#_Toc30030_WPSOffice_Level1)** **[1](#_Toc30030_WPSOffice_Level1)**

[2.1 消息事件接口](#_Toc30030_WPSOffice_Level2) [1](#_Toc30030_WPSOffice_Level2)

[2.2 方法接口](#_Toc23486_WPSOffice_Level2) [3](#_Toc23486_WPSOffice_Level2)

**[3.拓扑生成服务](#_Toc23486_WPSOffice_Level1)** **[3](#_Toc23486_WPSOffice_Level1)**

**[4.广播节点服务](#_Toc18148_WPSOffice_Level1)** **[3](#_Toc18148_WPSOffice_Level1)**

**[5.顶层节点在线服务](#_Toc7174_WPSOffice_Level1)** **[3](#_Toc7174_WPSOffice_Level1)**

**[6.Leader身份服务](#_Toc17169_WPSOffice_Level1)** **[3](#_Toc17169_WPSOffice_Level1)**

**[7.区块验证服务](#_Toc23564_WPSOffice_Level1)** **[3](#_Toc23564_WPSOffice_Level1)**

**[8.区块生成服务](#_Toc8824_WPSOffice_Level1)** **[3](#_Toc8824_WPSOffice_Level1)**

**[9.换届服务](#_Toc10248_WPSOffice_Level1)** **[3](#_Toc10248_WPSOffice_Level1)**

# 矿工服务

* 1. 消息事件接口
* 区块到来消息（eth现在就有的）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| chainHeadSub | Eth现有的 | 矿工服务 | type ChainHeadEvent struct{ Block \*types.Block } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 挖矿请求消息（存疑）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 挖矿请求消息到来 | Verify->p2p->?>矿工服务 | 矿工服务 | type MinerRequest struct{ header \*types.Header } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 身份变更消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| CA\_RoleUpdated | CA模块 | 矿工服务 | type RoleEvent struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 1. 方法接口

# 随机数生成服务

## 消息事件接口

* 随机种子生成请求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_SeedGenerate | 拓扑生成模块 | 随机数生成模块 | type RandomRequest struct {      Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 随机种子生成响应

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Random\_SeedGenerate | 随机数生成模块 | 拓扑生成模块 | type ElectionEvent struct {  Seed common.Hash // Hash represents the 32 byte Keccak256 hash of arbitrary data.  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 身份更新完成消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| CA\_RoleUpdated | CA模块 | 随机数服务模块  换届服务 | type RoleEvent struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

## 方法接口

* 公私钥生成方法

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 功能来自模块 | 功能使用模块 | 结构体 | 接口方式 | 备注 |
| 公私钥生成方法 | 网络组 | 算法组随机数服务 | 输入：高度(big.\*Int),type string ,data []byte  输出：无 | 直接调用 |  |

* 公私钥获取方法

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 功能来自模块 | 功能使用模块 | 结构体 | 接口方式 | 备注 |
| 公私钥获取方法 | 网络组 | 算法组随机数服务 | 输入：高度(big.\*Int),type string  输出：map  Key为交易地址，value为[]byte | 直接调用 |  |

# 拓扑生成服务

## 消息事件接口

## 接收：

* **网络生成请求：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_SelectTopNodes |  | 拓扑生成模块 |  | 订阅方式实现  直接传递结构体（不marshal  ） | 结构体包括：  类型（矿工还是验证者）  随机种子  区块高度  主节点列表 |

type SelectTopNodeEvent struct {

nodeList []election.NodeInfo

seed int64

    blockNumber \*big.Int

    updateType int

}

* **网络更新请求：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_UpdateTopNodes | 区块生成模块 | 拓扑生成模块 |  | 订阅方式实现  直接传递结构体（不marshal  ） | 结构体包括：  类型（矿工还是验证者）  更新类型（全更新还是快速更新）  区块高度  上线列表  离线列表  当前顶层节点列表  原始选举结果列表 |

type UpdateNetworkEvent struct {

    onlineList []election.NodeInfo

offlineList []election.NodeInfo

currentTopList []election.NodeInfo

selectResult []election.NodeInfo

    blockNumber \*big.Int

updateType int

role int

}

* **随机种子生成响应**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_RandSeed | 随机数生成模块 | 拓扑生成模块 | type ElectionEvent struct {  Seed common.Hash // Hash represents the 32 byte Keccak256 hash of arbitrary data.  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

## 发送

* 随机种子生成请求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
|  | 拓扑生成模块 | 随机数生成模块 | type RandomRequest struct {      Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

选举/更新结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_Result | 拓扑生成模块 | 换届模块  区块生成模块 |  | 订阅方式实现  直接传递结构体（不marshal  ） | 结构体包括：  类型（矿工还是验证者）  区块高度  一级节点列表  一级备份节点列表  二级备份节点列表 |

type ElectionResultEvent struct {

topList []election.NodeInfo

backupList1 []election.NodeInfo

bakcupList2 []election.NodeInfo

    blockNumber \*big.Int

    updateType int

}

type NodeInfo struct {

    TPS uint32 `json:"tps"`

    ID string `json:"id"`

    Wealth uint64 `json:"wealth"`

    OnlineTime uint64 `json:"online\_time"`

    Value float64 `json:"value"`

    Stock float64

}

## 方法接口

暂无

# 4.广播节点服务

# 5.顶层节点在线服务

## 5.1 消息事件接口

* 身份更新完成消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| CA\_RoleUpdated | CA模块 | 顶层节点在线服务 | type RoleUpdateMsg struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 拓扑变化请求消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Block\_TopoChangeReq | 区块生成 | 顶层节点在线服务 | type topoChangeReqMsg struct {   Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 拓扑更新结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Block\_TopoUpdateResult | 区块生成 | 顶层节点在线服务 | type topoChangeReqMsg struct {   Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* Leader变更消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| ·leader\_State | Leader身份计算服务 | 顶层节点在线服务 | type changeLeaderMsg struct {   Height \*big.Int//高度,可以为任何合法的高度值   ReelectTurn uint8//共识轮次    Leader common.address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

## 5.2 方法接口

* 获取顶层节点在线状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 获取顶层节点在线状态 | p2p | p2p ->顶层节点在线服务 |  | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

* 查询节点在线状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 查询节点在线状态 | p2p | p2p ->顶层节点在线服务 |  | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

* 查询节点原始身份

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 查询节点原始身份 | Ca | 顶层节点在线服务 |  | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

# Leader身份服务

* 身份更新完成消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| CA\_RoleUpdated | CA模块 | Leader身份计算服务，更换leader控制服务 | type RoleEvent struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 新区块header准备完毕消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 新区块header准备完毕消息 | 区块生成服务 | Leader身份计算服务，更换leader控制服务 | type newHeaderReady  struct {  Leader common.Address      Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* Blockchain区块插入消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Blockchain区块插入消息 | 拓扑生成模块 | Leader身份计算服务，更换leader控制服务 | type insertBlockMsg  struct {  Leader common.Address      Height \*big.Int//高度,可以为任何合法的高度值  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* Leader变更消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader变更消息 | Leader身份计算服务 |  | type insertBlockMsg  struct {  Leader common.Address  Height \*big.Int//高度,可以为任何合法的高度值  ConsensusTurn uint8 //共识轮次  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 更换leader通知消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_leaderChange | 拓扑生成模块 | Leader身份计算服务 | type leaderChangeMsg  struct {  leader common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证请求消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| P2P\_BlkVerifyRequest |  | 更换leader控制服务 | type BlockVerifyRequest struct {  Header types.Header//待验证header  TxsCode []uint32 // 交易列表  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| BlkVerify\_VerifyResult | 区块验证服务 | 更换leader控制服务 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 重选leader成功消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_reelectLeaderSuccess | 更换leader主节点服务 | 更换leader控制服务 | type reElectLeaderSuccEvent struct{  Leader  common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 启动重选leader主节点消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_leaderReelectStart | 更换leader控制服务 | 更换leader主节点服务 | type leaderReelectStartMsg struct {  Leader common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 停止重选leader主节点消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_leaderReelectStop | 更换leader控制服务 | 更换leader主节点服务 | type leaderReelectStopMsg struct{} | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 重选leader结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leade\_reelectLeaderResult |  | 更换leader主节点服务 | type reelectLeaderResultMsg struct {  voteResult vote  fromNode common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 启动重选leader从节点消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_followerReelectLeaderStart | 更换leader控制服务 | 更换leader从节点服务 | type followerReelectMsg  struct {  Leader common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 停止重选leader主节点消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_followerReelectLeaderStop | 更换leader控制服务 | 更换leader从节点服务 | type followerReelectMsg  struct {  Leader common.Address  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 重选leader请求消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_reelectLeaderReq | 更换leader控制服务 | 更换leader从节点服务 | type reelectLeaderRequestMsg struct {  fromNode common.Address  leaderDifference string //leader差值  timeStemp int64  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 重选leader结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_reelectLeaderResult | 更换leader控制服务 | 更换leader从节点服务 | type reelectLeaderRequestMsg struct {  fromNode common.Address  leaderDifference string //leader差值  timeStemp int64  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

# 7.区块验证服务

## 消息事件接口

* 区块验证请求消息(接受)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| P2P\_BlkVerifyRequest | 区块生成服务->p2p->net->p2p | 区块验证模块 | type BlockVerifyRequest struct {  Header types.Header//待验证header  TxsCode []uint32 // 交易列表  } | 订阅方式实现 | 区块生成服务调用p2p发送消息时，P2P也需要给自己的区块验证服务转发一份 |

* 区块验证投票消息(接受)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| P2P\_BlkVerifyVote | 区块验证模块->p2p->net->p2p | 区块验证模块 | type BlockVerifyVote struct {  SignHash common.Hash//签名的hash  Sign common.Signature //签名  } | 订阅方式实现 |  |

* 身份更新完成消息(接受)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| CA\_RoleUpdated | CA模块 | 区块验证模块 | type RoleUpdateMsg struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* Leader状态消息(接受)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader\_State | Leader身份服务 | 区块验证模块 | type LeaderStateMsg struct {  Leader common.Address  Number big.Int//  ReelectTurn uint8  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证结果消息(提供)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| BlkVerify\_VerifyResult | 区块验证模块 | 区块生成服务、Leader身份服务 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

## 方法接口

* 广播区块验证投票接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 接口提供模块 | 接口使用模块 | 结构体 | 接口方式 | 备注 |
| 广播区块验证投票消息接口 | 区块验证模块 | P2P->所有验证者 | type BlockVerifyVote struct {  SignHash common.Hash//签名的hash  Sign common.Signature //签名  } | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

* 广播区块验证结果接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 接口提供模块 | 接口使用模块 | 结构体 | 接口方式 | 备注 |
| 广播区块验证结果消息接口 | 区块验证模块 | P2P->所有矿工 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

* 交易验证接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 接口提供模块 | 接口使用模块 | 结构体 | 接口方式 | 备注 |
| 交易验证接口 | 区块验证模块 | 区块验证模块 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 方法接口直接调用。接口定义为  Func xxx(type uint8, data interface{}) error |  |

# 区块生成服务

## 8.1消息事件接口

* 挖矿结果验证请求消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 挖矿结果验证请求 | 矿工服务->p2p->net->p2p | 区块验证 | **type** MinerResult **struct** {  From common.Address  blockhash common.Hash  difficulty \*big.Int  nonce types.BlockNonce  Coinbase common.Address  MixDigest common.Hash  Signatures [] common.Signature } | 订阅方式实现 |  |

* 身份更新完成消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 身份更新完成消息 | CA模块 | 挖矿结果验证模块 | type RoleEvent struct {  RoleType uint8//uint8类型  Header \*common.Header//区块头  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证结果消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 区块验证结果消息 | 区块验证模块 | 区块生成服务 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 挖矿结果验证成功消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 挖矿结果验证成功消息 | 挖矿结果验证模块 | 区块生成服务、 | type BlockVerifyResultMsg struct {  Header types.Header//包含签名列表的header  TxsCode []uint32 // 交易列表  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* Leader状态消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Leader状态消息 | Leader身份服务 | 区块生成模块 | type LeaderStateMsg struct {  Leader common.Address  Number big.Int//  ReelectTurn uint8  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 启动Header生成服务消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 启动Header生成服务消息 | 区块生成服务 | 区块生成服务 | Number big.Int | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 启动Header生成服务消息 | 区块生成服务 | 区块生成服务 | Number big.Int | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 区块验证请求消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 区块验证请求 | 区块生成服务->p2p->net->p2p | 区块验证模块 | type BlockVerifyRequest struct {  Header types.Header//待验证header  TxsCode []uint32 // 交易列表  } | 订阅方式实现 | 区块生成服务发送消息时，P2P也需要给自己的区块验证服务转发一份 |

* 网络生成消息到来

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| ReElection\_TopologyGenerate | 换届服务 | 区块生成模块 | type TopologyGenerateEvent struct{  Type uint8  Topology []\*NodeInfo  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* **网络更新请求：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_UpdateTopNodes | 区块生成模块 | 拓扑生成模块 | type UpdateNetworkEvent struct {  Elect []common.Elect      onlineList []election.NodeInfo  offlineList []election.NodeInfo  topList []election.NodeInfo  backupList1 []election.NodeInfo  bakcupList2 []election.NodeInfo      blockNumber \*big.Int      updateType int  } | 订阅方式实现  直接传递结构体（不marshal  ） | 结构体包括：  选举结果  类型（矿工还是验证者）  区块高度  上线列表  离线列表  一级节点列表  一级备份节点列表  二级备份节点列表 |

## 8.2方法接口

* Nodeid转common addr接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Node2CommonAddr | ca模块 |  |  | 方法接口直接调用。接口定义为  func (self \*CA) Node2CommonAddr(id discover.NodeID) (common.Address,error) |  |

* 获取common addr对应的角色

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 接口名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Node2CommonAddr | ca模块 | 区块生成 |  | 方法接口直接调用。接口定义为  func (self \*CA) GetRole(common.Address) (uint32,error) {  return 0,nil } |  |

* 更新uptime接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 更新uptime接口 | ca模块 | 区块生成 | type xxxx struct {  uptime int64  addr common. Address //签名  } | 方法接口直接调用。接口定义为  Func xxx([]xxx) error |  |

* 获取编号对应的交易列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 获取编号对应的交易列表 | ca模块 | 区块生成 |  | 方法接口直接调用。接口定义为  Func xxx([]xxx) types.traction[] |  |

* 获取交易池所有交易

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| 获取编号对应的交易列表 | ca模块 | 区块生成 |  | 方法接口直接调用。接口定义为  Func xxx() types.traction[] |  |

* DPOS验证

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| DPOS验证共识 | Consensus | 区块生成 |  | VerifySignatures(signature []common.Signature) (bool,error) |  |

# 9.换届服务

## 9.1消息事件接口

* 区块到来消息（eth现在就有的）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| chainHeadSub | Eth现有的 | 换届服务 | type ChainHeadEvent struct{ Block \*types.Block } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 拓扑生成消息到来（接受）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| Election\_TopologyGenerate | 选举模块 | 换届服务 | type TopologyGenerateEvent struct{  Type uint8  Topology []\*NodeInfo  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

* 拓扑生成消息转发（提供）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 消息名称 | 消息来源模块 | 消息使用模块 | 结构体 | 接口方式 | 备注 |
| ReElection\_TopologyGenerate | 换届服务 | 区块生成模块 | type TopologyGenerateEvent struct{  Type uint8  Topology []\*NodeInfo  } | 订阅方式实现  直接传递结构体（不marshal  ） |  |

## 9.2方法接口